

基于扩展同步 Petri 网的 BPEL 建模

刘 贤, 李建华, 李 向, 陈 楠

(中南大学信息科学与工程学院, 长沙 410075)

摘 要: 使用业务流程执行语言(BPEL)描述复杂业务流程时容易出错。针对以上问题, 通过对同步 Petri 网进行扩展, 提出设计一个更适合 BPEL 建模的模型 SPN-NET, 给出各种 BPEL 结构化活动到 SPN-NET 的转换, 提出一种具有一般性的相似性化简算法。合理性验证和性能分析表明, 该模型简洁紧凑, 可理解性好。

关键词: 同步器; Petri 网; 业务流程执行语言

BPEL Modeling Based on Extended Synchronized Petri Net

LIU Xian, LI Jian-hua, LI Xiang, CHEN Nan

(School of Information Science and Engineering, Central South University, Changsha 410075, China)

【Abstract】 In order to solve the problem that it is prone to occur errors when Using the Business Process Execution Language(BPEL) to describe complex business process is error-prone. To solve this problem,, based on by extending synchronized Petri net, we this paper extended it in some terms and proposed designs a model named SPN-NET, which is more suitable for modeling BPEL, and translating translates a variety of BPEL structures into this the model was given, a. A general algorithm based on similar structure was is put forward.. Rationality validation A verification of the soundness and performance analysis show that the model is simple , compact and comprehensible.

【Key words】 synchronizer; Petri net; Business Process Execution Language(BPEL)

1 概述

目前, 业务流程执行语言(Business Process Execution Language, BPEL)已成为执行过程定义的标准。由于商业逻辑的复杂性, 使用 BPEL 组合 Web 服务容易出错, 因此需要采用形式化方法来建模、分析、验证 BPEL 流程。很多学者利用 Petri 网^[1]和有色 Petri 网^[2]对 BPEL 建模, 但复杂流程容易导致模型的爆炸性增长。也有用同步 Petri 网^[3]对 BPEL 建模的, 相比 Petri 网, 同步 Petri 网其能以较简洁的方式进行描述, 但同步网的语义层采用库所表示变迁的输入输出参数, 当输入输出参数增多时, 同样会导致模型的爆炸。本文在同步网的基础上引入了一些有色 Petri 网的概念, 针对 BPEL 的特点, 在不改变同步网基本性质的前提下, 提出了一种更适合 BPEL 建模的 SPN-NET 模型。

2 SPN-NET 模型

2.1 模型形式化表示

SPN-NET 可以用十一元组 $(T, P, K, W, A, Lp, Lk, G, R, C, \Sigma)$ 表示, 其中:

(1) T 是变迁, 分为瞬时变迁、基本变迁和分层可设置(HM)变迁。

(2) P 是库所, 表示同步器。

(3) K 表示库所容量。 $K(p)=m_1 \times m_2$, 其中, m_1 、 m_2 分别表示前置和后置分支数, 分支不包括循环分支。

(4) W 是弧上的权。 $W(t, p)=m_2$, $W(p, t)=m_1$ 。注意, 由于颜色 token 的引入, 因此这里的 m_1 和 m_2 是指分支数, 而不是激活分支数。

(5) A 是弧的集合, $P \cap T = P \cap A = T \cap A = \emptyset$ 。

(6) Lp 为循环线, 用来表示循环, 与普通弧线不同的是, $W(Lp)=K(p)$, 其中, $K(p)$ 为 Lp 所连接库所的容量, 其限制条件为: 循环起点的前驱和后继的变迁集合等于终点的前驱

和后继的变迁集合。

(7) Lk 称为链接, 在 BPEL 中表示活动间的同步。

(8) G 是哨函数, 满足条件: $\forall t \in T: [type(G(t))=B \text{ type}(Var(G(t))) \subseteq \Sigma]$ 。在本模型中哨函数都为 $token.live=true$ 。对于 $token.live=false$ 的变迁仅是向后(不包括循环)传递 token, 但在 BPEL 的 flow 元素中, 若设置 $suppressJoinFailure=no$, 则抛出异常。这能很好地解决 flow 元素中的 link 问题和死路径消除(Dead Path Elimination, DPE)问题。

(9) R 是变迁条件, 定义弧 $a: A$ 到布尔表达式的映射, 满足条件: $\forall a: A: [type(R(a))=B \text{ type}(var(R(a))) \subseteq \Sigma]$, 缺省时为 true, $R(a)$ 决定 $token.live$ 的值。

(10) Σ 是颜色集合。

(11) C 是颜色函数, 是 P 到 Σ 的映射。

2.2 变迁发生条件

变迁发生条件包括:

(1) 除循环结构外, 其他情况下每个变迁至多只能发生一次。

(2) 只有在 $M(p)=K(p)$ 时才能同步授权给 T_2 。

(3) t 在 M 有发生权的条件是:

$\forall p \in t: M(p) \geq W(p, t) \quad \forall p \in t: M(p) + W(t, p) \leq K(p)$

标识的变化规律为:

$\forall p \in P: M_2(p) = M_1(p) - W(p, t) + W(t, p)$

2.3 BPEL 到模型的转换

BPEL 提供了 2 种类型的活动: 原子活动和结构化活动。原子活动包括 $\langle receive \rangle$ 、 $\langle reply \rangle$ 、 $\langle invoke \rangle$ 、 $\langle assign \rangle$ 等, 其可以简单地用一个变迁来表示, 而用库所表示活动执行的

作者简介: 刘 贤(1984 -), 男, 硕士研究生, 主研方向: Petri 网, 服务计算; 李建华, 教授; 李 向、陈 楠, 硕士

收稿日期: 2010-06-30 **E-mail:** liuxiaif@163.com

前提条件和后续状态,如图 1(a)所示。结构化活动包括 <sequence>、<while>、<flow>、<switch>和<pick>等元素。各种结构化活动到 SPN-NET 的转换如下:(1)<sequence>表示多个任务顺序执行,顺序活动到模型的映射到模型如图 1(b)所示。(2)<while>表示某个任务多次重复执行。循环活动映射到模型的映射如图 1(c)所示。 $R(a)=C$ 是一个循环条件,用来确定是否执行循环体 A1。(3)<switch>表示从多个任务中选择执行,选择活动到模型的映射到模型如图 1(d)所示, $R(a_i)=C_i(i=1,2,\dots,n)$ 是选择条件。(4)<pick>与 switch 结构类似,本文不再赘述。(5)<flow>表示多个任务并行执行。并行活动到模型的映射到模型如图 1(e)所示。(6)<link>表示同步行为。含有 link 的 flow 结构如图 1(f)所示。图 1 中,粗实线表示链接;C 表示链接条件;join 表示对连接条件的判断,如果有多个活动到 A_n 的连接,则同样增加从其他活动到 join 的连接链接。当所有到 A_n 的连接条件为 true 时, A_n 正常执行,否则,由 suppressJoinFailure 属性来决定。

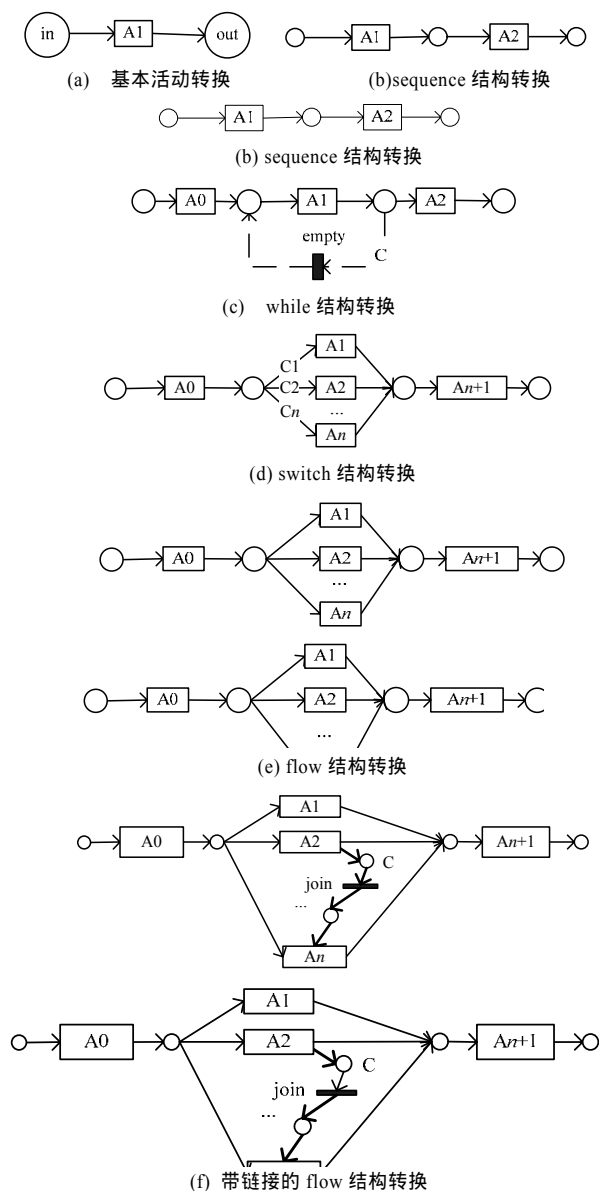


图 1 BPEL 结构化活动到 SPN-NET 的转换

相比其他 Petri 网,本模型具有以下优点:(1)库所颜色的引入使模型含义更加丰富,同时简化了流程状态转移规则。

(2)模型显得更加紧凑简洁,可理解性更好。(3)更好地解决了 BPEL 的 flow 元素中的 link 问题和死路径消除问题。(4)变迁条件的引入使模型更符合 BPEL 的特点和用户的使用习惯。

3 模型合理性验证和性能分析

3.1 模型结构验证

文献[3]归纳了同步网验证的 8 个化简规则,如果同步网能化简为一个库所,则它是畅通的,反之并不能判定它是否畅通。所以,本文在普通 Petri 网可达图的基础上给出了此模型的可达图分析方法。由于 Petri 网基于交错发生语义,而本模型是基于同步发生语义,因此做了 2 点改动:(1)同步器同步触发的可选变迁在可达图中以并发的关系出现。(2)达到容量的同步器进行状态转移,多个同步器是交错的,即一次状态转移只有一个同步器同步触发可选变迁,如果不同同步器的可选变迁之间有交集,则并行转移。篇幅所限,本文不给出具体的算法描述。

结论:若流程 Σ 的可达图存在且叶节点都是结束状态,则 Σ 是畅通的,反之亦然。若流程 Σ 的可达图存在且所有弧标识的集合等于 T,则 Σ 没有死变迁;若流程 Σ 的可达图存在且包含有向圈,则 Σ 中存在循环。

3.2 模型性能分析

目前模型性能分析方法主要有基于马尔可夫链(MC)和基于等价化简。本文采用基于马尔可夫链的性能分析方法,步骤如下:

(1)服务执行速率和消息传输速率服从指数分布,它同构于一个连续时间马尔可夫链 MC。

(2)模型的状态可达图同构于一个 MC 的状态空间。

(3)获得 MC 转移速率矩阵的参数,与一般 Petri 网的可达图不同的是,对于并发的变迁,这个参数需要通过等价转换^[4]才能得到。

(4)计算出 MC 的每个标识在稳定状态下的稳定概率,在此基础上分析库所 token 概率密度和变迁的利用率等,得出组合服务的平均实施时间、性能瓶颈和吞吐量等。

用此方法对文献[5]的 WS SodaSys 组合服务进行分析,得出的结果与服务实际执行时间相差 8%左右。所以,认为此方法是有效的。

4 实例分析

下面以一个旅游行程安排为例进行分析。首先获取旅程车行时间,根据时间决定交通方式,然后获取景点信息,确定游玩路线,获取酒店信息,在预订酒店前需要和与获取旅游信息同步,最后输出最终方案。实例的 BPEL 描述如下:

```
<sequence>
  <invoke name=""ws1"></invoke>
  <switch>
    <case condition=""t>8">ws2</case>
    <case condition=""t<=8">ws3</case>
  </switch>
  <flow suppressJoinFailure=""yes">
    <links>
      <link name=""link1"" />
    </links>
    <sequence>
      <invoke name=""ws4">
        <source link=""link1""
          transitionCondition=""C""/></invoke>
```

```
<invoke name=""ws6"></invoke>
</sequence>
<sequence>
<invoke name=""ws5"></invoke>
<invoke name=""ws7">
<target link="link1" /></invoke>
</sequence>
</flow>
<invoke name=""ws8"></invoke>
</sequence>
```

旅游行程安排流程如图 2 所示，其中，ws1 为获取旅行
行车时间；ws2 为订购飞机票；ws3 为订购火车票；ws4 为查
询景点相关信息；ws5 为查询酒店相关信息；ws6 为确定旅
游路线；ws7 为预定预订酒店；ws8 为方案显示和输出。

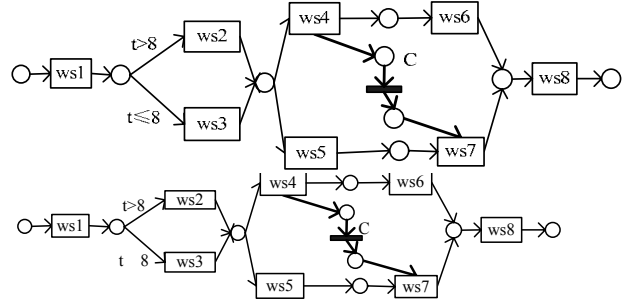


图 2 旅游行程安排流程图

图 3 是图 2 的可达图，从中可以看出，图 2 的流程是畅
通的，没有死变迁，不存在循环。

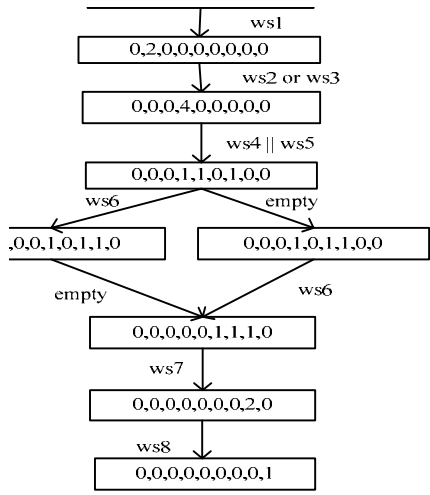
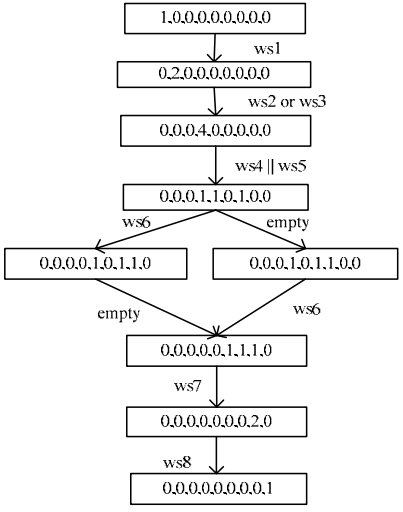


图 3 旅游行程安排流程的可达图

5 模型进一步简化

本节提出了一种基于结构相似性的 Petri 网化简算法。下
面结合图 4 的模型来说明该算法。

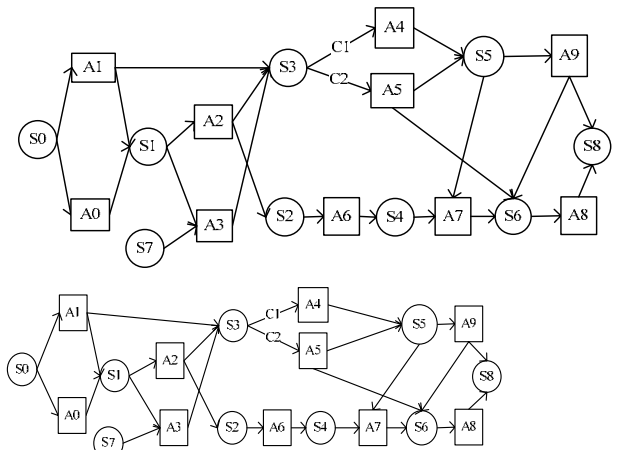


图 4 相似性化简示例模型

算法使用了 2 个哈希表，一个哈希表包含所有库所及它
其前置和后置变迁，另一个用来装所有变迁以及它及其前置
和后置库所。表 1 中给出了库所的哈希表，哈希函数 $H(n:m)$ ，
其中， n 表示前置变迁数目； m 表示后置变迁数目。此函数
返回所有具有相同 n 和 m 值的库所列表，每个库所用形如
 $S(A1,A2:A3,A4)$ 来表示，其中， $A1$ 、 $A2$ 表示前置变迁， $A3$ 、
 $A4$ 表示后置变迁。表 2 给出了变迁的哈希表。

表 1 库所 hash 哈希表

$HH(nn,mm)$	库所列表($nn:mm$ 变迁)
$HH(0,1)$	$S7(A3)$
$HH(0,2)$	$S0(A0,A1)$
$HH(2,0)$	$S8(A8,A9)$
$HH(1,1)$	$S2(A2,A6), S4(A6,A7)$
$HH(2,2)$	$S1(A0,A1:A2,A3), S5(A4,A5:A7,A9)$
$HH(3,1)$	$S6(A5,A7,A9:A8)$
$HH(3,2)$	$S3(A1,A2,A3:A4,A5)$

表 2 变迁 hash 哈希表

$HH(nn,mm)$	变迁列表($nn:mm$ 库所)
$HH(1,1)$	$A0(S0:S1), A4(S3:S5), A6(S2:S4), A8(S6:S8)$
$HH(1,2)$	$A1(S0:S1,S3), A2(S1:S2,S3), A9(S5:S6,S8), A5(S3:S5, S6)$
$HH(2,1)$	$A7(S4,S5:S6), A3(S1,S7:S3)$

建立了这 2 个表之后,对表 1 中的每个 $HH(n:m)$ 按照从复杂到简单的顺序进行分析。

(1)因为 $HH(3,2)$ 、 $HH(3,1)$ 只有一个库所,所以不用考虑,对 $HH(2,2)$ 进行分析。

(2)对于由于表 1 中的一个哈希函数,它的其所有库所都有相同的输入和输出变迁数,由此因此可以把它们划为同一组($S1, S5$)。

(3)对找出的相同组,先考虑它的输入变迁,这需要用到表 2, $S1=HH(1,1)$, $HH(1,2)$, $S5=HH(1,1)$, $HH(1,2)$, 因此,这个组还是($S1, S5$)。考虑它的输出变迁,在表 2 中, $S1=HH(1,2)$, $HH(2,1)$, $S5=HH(2,1)$, $HH(1,2)$, 所以,分组仍然是($S1, S5$)。

(4)判断组内是否有重叠变迁,如果有,则留下其中一个, ($S1, S5$)仍然为同一组。

找出一个分组后进行替换:对于组内的每个库所,把它以及与其相连的变迁移出来,用 HM 变迁替换,在 HM 变迁中通过虚拟的库所(虚线圆圈)与外部流程相连,如图 5 所示。注意,虚拟库所实际是不存在的,不属于子结构,所以,含有虚拟库所的结构不一定相同。当 HM 变迁使能时,它周围的库所 token 便会进入到 HM 变迁对应的虚拟库所中,当然也包括弧类型、条件、变迁信息以及其他的资源。替换完后重复以上步骤,直到没有相同结构为止。

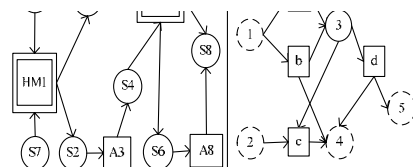
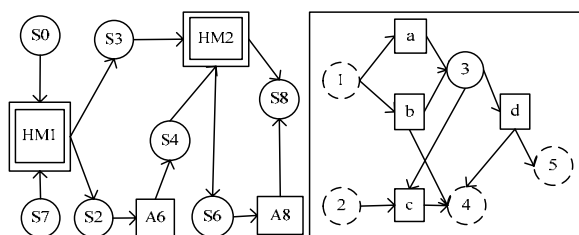


图 5 相似性化简后的模型

6 结束语

本文基于 SPN-NET 对 BPEL 进行建模,建立的模型简洁易懂。但诸如建模工具的实现、数据竞争检测分析等都是有待进一步研究的工作。

参考文献

- [1] 袁崇义. Petri 网原理与应用[M]. 北京: 电子工业出版社, 2005.
- [2] 门 鹏, 段振华. 基于着色 Petri 网的 BPEL 建模与验证[J]. 西北大学学报, 2007, 37(6): 986-990.
- [3] Dun Haiqiang, Xu Haiying. Transformation of BPEL Processes to Petri Nets[C]//Proc. of the 2nd International Symposium on Theoretical Aspects of Software Engineering. Nanjing, China: [s. n.], 2008: 166-173.
- [4] 龙土工, 罗文俊. 随机 Petri 网性能等价化简与分析方法研究[J]. 计算机工程, 2006, 32(12): 27-28, 36.
- [5] Henrique J. A Framework for the Performance Analysis of Web Services Orchestrated with BPEL4WS[C]//Proc. of the 4th International Conference on Internet and Web Application and Service. Venice, Italy: [s. n.], 2009: 363--369.

编辑 张 帆