基于Petri网的面向测试的工作流系统建模方法

郑长友，刘晓明，姚 奕，任正平

(解放军理工大学指挥自动化学院 南京 210007)

【摘要】提出了一种基于Petri网的I/O_WF_Net模型。该模型将工作流中的活动抽象为Petri网中的迁移，将每个活动的输入输出抽象为Petri网中的库所，便于以后应用其生成测试用例。介绍了对工作流中各种组成部分及不同结构的I/O_WF_Net模型建模方法，给出了将带有输入输出约束的工作流网转化为I/O_WF_Net模型的算法及转化后模型的化简方法。

关 键 词 Petri网; 工作流; 工作流建模; 工作流测试

中图分类号 TP311 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.01.020

Modeling Approach for Workflow Testing Based on Petri Nets

ZHENG Chang-you, LIU Xiao-ming, YAO Yi, and REN Zheng-ping

(Institute of Command Automation, PLA University of Science and Technology Nanjing 210007)

Abstract An I/O_WF_Net model based on Petri nets is proposed in this paper. In the I/O_WF_Net model, the activities of a workflow are abstracted as transitions and the inputs and outputs of an activity are abstracted as places of a Petri net, so the model is convenient for test cases generating. The modeling method of the components and structures of a workflow are described, an algorithm that transforms a workflow net constrained by inputs and outputs into the I/O_WF_Net model and the corresponding simplifying method are given.

Key words Petri Nets; workflow; workflow modeling; workflow testing

工作流系统建模技术已经发展多年，并取得了诸多成果[1-5]。但这些研究大都集中在工作流的任务调度、正确性验证或性能分析上[3-5]，缺乏真正面向测试的建模技术。目前，对工作流系统测试所使用的测试用例大都依靠测试人员自身经验给出，这些用例也大都需要人工执行。这种现状带来以下两方面的问题：1) 降低了软件测试的成熟度和充分性；2) 无法保证系统的性能及可靠性。因此，需要对工作流测试进行有针对性地深入研究。

文献[6]分析了工作流系统的特性和其中可能出现的几类缺陷，提出了一种工作流自动动态测试的框架，该框架的定义以BNF(backus-naur form)形式给出，对于各种工作流定义语言均可使用。但由于BNF定义比较复杂，并且语义也较难理解，因此该方法不便于广泛的应用。

文献[7]总结了应用MVC设计模式的web应用中缺陷的类型，并分别提出了基于状态和基于代码的用于测试此类web工作流应用的抽象模型，但是该模型只给出了抽象策略，具体应用方法和详细测试技术并没有给出，实用性不够强。

除了上述的研究之外，文献[8]开发了一种自动测试工作流系统的工具，该工具通过量化的方式评估工作流系统的性能。文献[9]给出了一种基于工作流的自动测试数据分析方法，该方法分析测试执行完成后得到的相应数据，而不是针对工作流测试本身的技术。

由于Petri网是一种图形化的建模语言，并且具有严格的数学语义，加之目前对其分析技术的研究已经颇为丰富和成熟，因此很适用于对工作流进行建模[10-11]。

本文在Petri网的基础上提出了一种I/O_WF_Net模型，该模型将工作流中的活动抽象为Petri网中的迁移，将每个活动的输入输出抽象为Petri网中的库所，介绍了对工作流的各种组成部分及不同结构进行建模的方法，给出了将带有输入输出约束的工作流网转化为I/O_WF_Net模型的转化算法及转化后模型的化简方法，通过一个实例介绍了该建模方法的全过程，并证明了该方法的有效性。

收稿日期：2012－07－09；修回日期：2012－10－25

基金项目：国家863项目(2009AA01Z402)；中国博士后科学基金面上项目(20110491843)；江苏省自然科学基金(BK2012059,BK2012060)

作者简介：郑长友(1986－)，男，博士生，主要从事软件测试、可信软件方面的研究.

1 基于Petri网的带输入输出约束的工作流网

1.1 带输入输出约束的工作流

对Petri网的具体介绍可以参见文献[12-14]，由于其不是本文的重点，在此不再赘述。

定义 1 六元组<Activity,Input,Output,Relation, f_{AI},f_{AO} >表示带输入输出约束的工作流，其中：

1) $\text{Activity}=\{\text{activity}_1,\text{activity}_2,\cdots,\text{activity}_k\}(k\geqslant 1)$ 表示工作流的活动集合。

2) $\text{Input}=\{\text{input}_1,\text{input}_2,\cdots,\text{input}_m\}(m\geqslant 1)$ 表示输入元素的集合。

3) $\text{Output}=\{\text{output}_1,\text{output}_2,\cdots,\text{output}_n\}$ $(n\geqslant 1)$ 表示输出元素的集合。

4) $\text{Relation}\subseteq(\text{Activity}\times\text{Activity},\text{Type})$ 表示工作流中的关系集合，其中，$\text{Type}\subseteq$ {sequence, and-join, or-join, and-split, or-split} 表示工作流中前后两个活动的关系类型。对集合 Activity 中任意的 $\text{activity}_1,\text{activity}_2$，如果 $(\text{activity}_1,\text{activity}_2)\in$ Relation，那么 activity_1 是 activity_2 的前驱活动，activity_2 是 activity_1 的后续活动。Type定义了前后两个活动关系的类型：sequence表示“顺序”类型，即前驱活动执行后，后续活动才能执行；and-join和or-join分别表示“与聚合”和“或聚合”类型，这两种类型关系的前驱活动可能有多个，不同之处在于and-join类型下只有所有前驱活动均执行完毕后，后续活动才能执行，而or-join类型下无需等待所有前驱活动执行完毕，只要有一个(或几个)前驱活动执行完毕，后续活动即可以执行；and-split和or-split分别表示“与分支”和“或分支”类型，该类型的后续活动有多个，它们之间的不同点是and-split类型下只要前驱活动执行完毕，所有后续活动均可以执行，而or-split类型下前驱活动执行完毕后，在所有后续活动中根据一定条件选择一个(或几个)进行执行。

1) $f_{AI}:\text{Activity}\rightarrow\rho(\text{Input})$ 表示工作流中某一活动到输入元素的映射，其中，$\rho(\text{Input})$ 表示输入元素的幂集，如果 $f_{AI}(\text{activity}_1)=\text{input}_1$ 成立，则表明活动 activity_1 的执行需要输入元素 input_1。

2) $f_{AO}:\text{Activity}\rightarrow\rho(\text{Output})$ 表示工作流中某一活动到输出元素的映射，其中，$\rho(\text{Output})$ 表示输出元素的幂集，如果 $f_{AO}(\text{activity}_1)=\text{Output}_1$ 成立，则表明活动 activity_1 执行后产生输出 output_1。

表1给出了一个带输入输出约束工作流的例子。从该表中可以看出，工作流由9个活动组成，表中的第2列和第3列分别给出了各个活动的输入和输出内容，各活动的前驱活动和相应活动间的关系类型在该表的第4和第5列给出。

表1 一个带输入输出约束的工作流

Activity	Inputs	Outputs	Pre-activities	Relation type
A_1	p_1	p_2	$\varnothing$	$\varnothing$
A_2	p_2	p_3	A_1	and-split
A_3	p_2	p_4	A_1	and-split
A_4	p_5	p_8	A_2	or-split
A_5	p_5	p_9	A_2	or-split
A_6	p_6	p_{10}	A_3	and-split
A_7	p_7	p_{10}	A_3	and-split
A_8	p_{11}	p_{13}	A_4, A_5	or-join
A_9	p_{12}	p_{14}	A_6, A_7	and-join

1.2 带输入输出约束工作流的Petri网模型

定义 2 $\Sigma=(P,T,F,M_0)$ 是一个I/O_WF_NET，当且仅当满足如下条件：

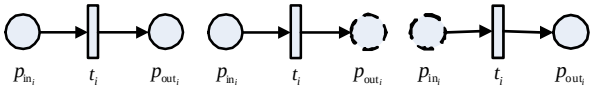
1) (P,T,F,M_0) 是一个Petri网；2) T为工作流中的活动集合；3) $P=P_{\text{in}}\cup P_{\text{out}}$ 且 $P_{\text{in}}\cap P_{\text{out}}=\varnothing$，其中 P_{in} 为输入类库所，P_{out} 表示输出类库所；4) $\forall t\in T$，$p_{\text{in}}\in P_{\text{in}}$，活动 t 的执行需要输入 p_{in} 当且仅当 $p_{\text{in}}\subseteq {}^{\bullet}t$；5) $\forall t\in T$，$p_{\text{out}}\in P_{\text{out}}$，活动$t$的执行产生输出 p_{out} 当且仅当 $p_{\text{out}}\subseteq t^{\bullet}$。

2 一种面向I/O_WF_NET的建模方法

本节中将给出基于I/O_WF_NET模型的工作流建模方法，本文假设工作流中的每个活动均只需要一个输入内容并产生一个输出结果。

2.1 单一活动建模

由于工作流是由多个不同活动相互连接组成，因此对工作流建模前，先对其基本组成成分——单一的活动进行建模，如图1所示。



a. 有输入和输出　　b. 有输入无输出　　c. 无输入有输出

图1 单个活动建模

2.1.1 既有输入又有输出的活动建模

既有输入又有输出的活动在工作流中最为常见，该类活动在I/O_WF_NET模型中可表示为两个库所和一个迁移的形式，如图1a所示。

图中，t_i 表示活动 i，p_{in_i}、p_{out_i} 分别表示活动 i 的输入和输出。如果 p_{in_i} 中含有token，则表示活动 i 的输入条件满足，该活动可以执行；如果 p_{out_i} 中含有token，则表示活动 i 经过执行后输出了 p_{out_i}。形式化上，活动 t_i 可以表示为 $[p_{\text{in}_i},t_i,p_{\text{out}_i}]$。

2.1.2 有输入无输出的活动建模

如果活动 t_i 执行后不产生任何输出，则在建模时为其增加一个虚拟输出库所 p_{out_i}，该库所不表示任何实际的物理意义，并用虚线圆圈的形式表示，如图1b所示。

2.1.3 无输入有输出的活动建模

如果活动 t_i 无需任何输入便可执行，只是在工作流执行时与其前后的活动有先后的顺序关系，在建模时为其增加一个虚拟输入库所 p_{in_i}，该库所不表示任何实际的物理意义，并用虚线圆圈的形式表示，如图1c所示。

2.2 工作流基本关系建模

如果 $(t_i,t_j)\in \text{Relation}$，即活动 $[p_{\text{in}_j},t_j,p_{\text{out}_j}]$ 是活动 $[p_{\text{in}_i},t_i,p_{\text{out}_i}]$ 的后续活动，则在建模时，需根据两个活动的关系类型进行建模。

2.2.1 sequence类型关系建模

如果活动 t_i、t_j 间关系为sequence类型，那么建模时需要在活动 t_i、t_j 之间增加一个新的虚拟迁移 t_{ij}，且 ${}^\bullet t_{ij}=p_{\text{out}_i}$，$t_{ij}^\bullet=p_{\text{in}_j}$，如图2所示。该迁移无任何实际的物理意义，只是起到连接活动 t_i 与 t_j 的作用。

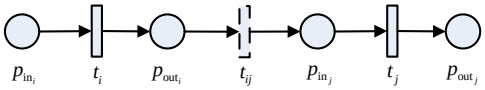


图2　sequence类型关系建模

2.2.2 and-join类型关系建模

如果活动 t_i、t_j 间关系为and-join类型，那么建模时首先查看 ${}^\bullet p_{\text{in}_j}$ 是否等于∅：若 ${}^\bullet p_{\text{in}_j}=\varnothing$，则需要在活动 t_i、t_j 之间增加一个新的虚拟迁移 t_{ij}，且 ${}^\bullet t_{ij}=p_{\text{out}_i}$，$t_{ij}^\bullet={}^\bullet p_{\text{in}_j}$，此时得到的结果与2.2.1节中结果一致；若 ${}^\bullet p_{\text{in}_j}\neq\varnothing$，则表明已经有与活动 t_j 存在and-join类型关系的活动 t_k 在建模时增加过虚拟活动 t_{kj} 了，此时直接令 $p_{\text{out}_i}^\bullet={}^\bullet p_{\text{in}_j}$ 即可，如图3所示。

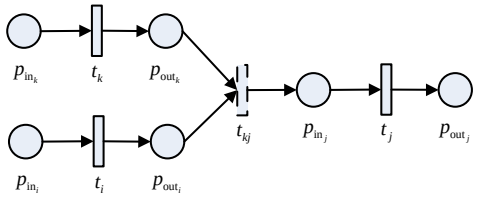


图3　and-join类型关系建模

2.2.3 or-join类型关系建模

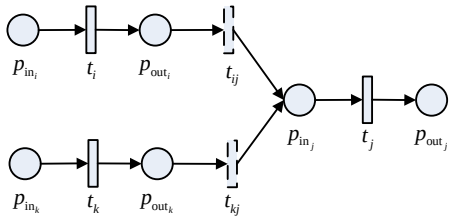


图4　or-join类型关系建模

如果活动 t_i、t_j 间关系为or-join类型，那么建模时与sequence类型关系一样，直接在活动 t_i、t_j 之间增加一个新的虚拟迁移 t_{ij}，令 ${}^\bullet t_{ij}=p_{\text{out}_i}$，$t_{ij}^\bullet=p_{\text{in}_j}$ 即可，对or-join类型的关系建模后如图4所示。

2.2.4 and-split类型关系建模

如果活动 t_i、t_j 间关系为and-split类型，那么建模时首先查看 $p_{\text{out}_i}^\bullet$ 是否等于∅：若 $p_{\text{out}_i}^\bullet=\varnothing$，则需要在活动 t_i、t_j 之间增加一个新的虚拟迁移 t_{ij}，且 ${}^\bullet t_{ij}=p_{\text{out}_i}$，$t_{ij}^\bullet=p_{\text{in}_j}$，此时得到的结果与2.2.1节中结果一致；若 $p_{\text{out}_i}^\bullet\neq\varnothing$，则表明已经有与活动 t_i 存在and-split类型关系的活动 t_k 在建模时增加过虚拟活动 t_{ik} 了，此时直接令 ${}^\bullet p_{\text{in}_j}=p_{\text{out}_i}^\bullet$ 即可，如图5所示。

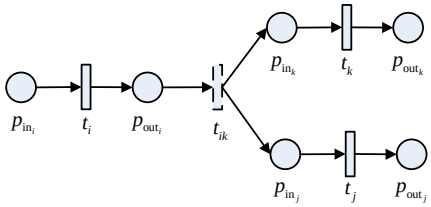


图5　and-split类型关系建模

2.2.5 or-split类型关系建模

如果活动 t_i、t_j 间关系为or-split类型，那么建模时与sequence类型关系一样，直接在活动 t_i、t_j 之间增加一个新的虚拟迁移 t_{ij}，令 ${}^\bullet t_{ij}=p_{\text{out}_i}$，$t_{ij}^\bullet=p_{\text{in}_j}$ 即可，对or-join类型的关系建模后如图6所示。

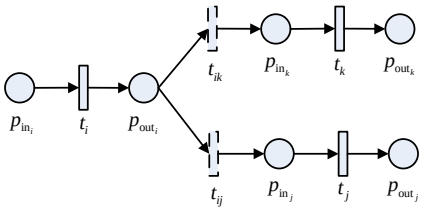
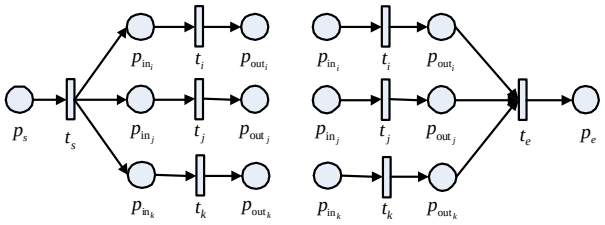


图6　or-split类型关系建模

2.3 起始与终止活动建模

起始与终止活动建模如图7所示。



a. 起始活动 b. 终止活动

图7 起始与终止活动建模

2.3.1 起始活动建模

如果活动 t_i 没有前驱活动，即 ${}^{\bullet}p_{\mathrm{in}_i}=\varnothing$，那么在建模时，为这些活动统一增加一个前驱活动 t_{s}，使得 $p_{\mathrm{in}_i}\subseteq t_{\mathrm{s}}^{\bullet}$。另外，为保证Petri网的基本特性，再增加一个起始库所 p_{s}，使得 ${}^{\bullet}t_{\mathrm{s}}=p_{\mathrm{s}}$，如图7a所示。

2.3.2 终止活动建模

如果活动 t_i 没有后续活动，即 $p_{\mathrm{out}_i}^{\bullet}=\varnothing$，那么在建模时，为这些活动统一增加一个后续活动 t_{e}，命名为终止活动，使得 $p_{\mathrm{out}_i}\subseteq {}^{\bullet}t_{\mathrm{s}}$。另外，为保证Petri网的基本特性，再增加一个终止库所 p_{e}，使得 $t_{\mathrm{e}}^{\bullet}=p_{\mathrm{e}}$，如图7b所示。

2.4 初始标记

I/O_WF_NET模型 $\Sigma=(P,T,F,M_0)$ 的初始标记满足如下的条件：

$$M_0(p)=\begin{cases}1 & p=p_s\\ 0 & \text{其他}\end{cases} \tag{1}$$

2.5 转换算法

在上文阐述的I/O_WF_NET模型建模方法的基础上，算法1给出了将带输入输出约束的工作流网转化为I/O_WF_NET模型的具体方法和过程。

算法 1 算法输入为Workflow=<Activity, Input, Output, Relation, f_{AI}, f_{AO}>，算法输出为 $\Sigma=(P,T,F,M_0)$。

算法过程：

begin

$P\leftarrow\varnothing, T\leftarrow\varnothing, F\leftarrow\varnothing, M_0\leftarrow\varnothing$

$T\leftarrow$ Activity

for $i=1$ to $|T|$ do

$p_{\mathrm{in}_i}\leftarrow F_{\mathrm{AI}}(t_i)$， $p_{\mathrm{out}_i}\leftarrow F_{\mathrm{AO}}(t_i)$

$P\leftarrow P\cup\{p_{\mathrm{in}_i}, p_{\mathrm{out}_i}\}$

$F\leftarrow F\cup\{(p_{\mathrm{in}_i},t_i),(t_i,p_{\mathrm{out}_i})\}$

end for

$\forall t_i,t_j\in T(1\leqslant i,j\leqslant|T|)$ if $t_i,t_j\in$ Relation then

Swich Relation.Type

Case and-join:

If ${}^{\bullet}p_{\mathrm{in}_j}=\varnothing$ then

$T\leftarrow T\cup t_{ij}$

$F\leftarrow F\cup\{(p_{\mathrm{out}_i},t_{ij}),(t_{ij},p_{\mathrm{in}_j})\}$

Else

$F\leftarrow F\cup\{(p_{\mathrm{out}_i},{}^{\bullet}p_{\mathrm{in}_j})\}$

Case and-split:

If $p_{\mathrm{out}_i}^{\bullet}=\varnothing$ then

$T\leftarrow T\cup t_{ij}$

$F\leftarrow F\cup\{(p_{\mathrm{out}_i},t_{ij}),(t_{ij},p_{\mathrm{in}_j})\}$

Else

$F\leftarrow F\cup\{(p_{\mathrm{out}_i}^{\bullet},p_{\mathrm{in}_j})\}$

Default:

$T\leftarrow T\cup t_{ij}$

$F\leftarrow F\cup\{(p_{\mathrm{out}_i},t_{ij}),(t_{ij},p_{\mathrm{in}_j})\}$

End Swich

$T_{\mathrm{s}}\leftarrow\{t_i\mid t_i\in T,(p_{\mathrm{in}_i},t_i)\in F\wedge{}^{\bullet}p_{\mathrm{in}_i}=\varnothing\}$

$P\leftarrow P\cup p_{\mathrm{s}}$， $T\leftarrow T\cup t_{\mathrm{s}}$， $F\leftarrow F\cup\{(p_{\mathrm{s}},t_{\mathrm{s}})\}$

for each $t_i\in T_{\mathrm{s}}$ do

$F\leftarrow F\cup\{(t_{\mathrm{s}},p_{\mathrm{in}_i})\}$

end for

$T_{\mathrm{e}}\leftarrow\{t_i\mid t_i\in T,(t_i,p_{\mathrm{out}_i})\in F\wedge p_{\mathrm{out}_i}=\varnothing\}$

$P\leftarrow P\cup p_{\mathrm{e}}$， $T\leftarrow T\cup t_{\mathrm{e}}$， $F\leftarrow F\cup\{(t_{\mathrm{e}},p_{\mathrm{e}})\}$

for each $t_i\in T_{\mathrm{e}}$ do

$F\leftarrow F\cup\{(p_{\mathrm{out}_i},t_{\mathrm{e}})\}$

end for

$M_0(p_{\mathrm{s}})\leftarrow 1$

输出 $\Sigma=(P,T,F,M_0)$

end

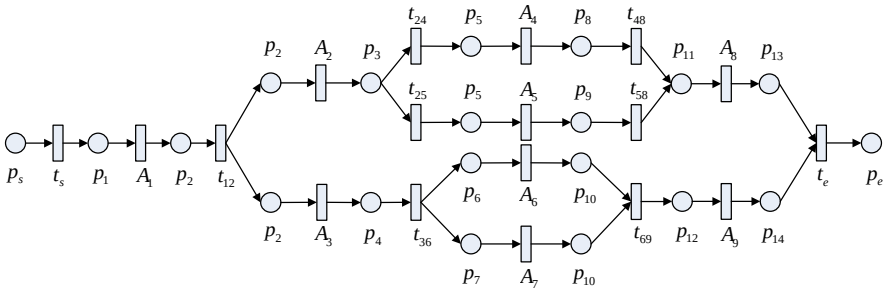


图8　应用算法1对表1描述的工作流建模结果

算法的时间消耗主要花费在几个循环上，其中第一个for循环的时间消耗是$O(|T|)$；在第二步中加入前后活动的关系时，其所需时间耗费是$O(|\text{Relation}|)=O(|T|\times|T|)=O(|T|^2)$的；由于模型中起始活动和结束活动的个数一定是不大于$|T|$的，因此对起始活动和结束活动建模，其所需时间消耗不大于$O(|T|)$；最后对整个模型进行初始标记其所需时间是$O(1)$。综上所述，整个算法的时间复杂性是$O(|T|^2)$。图8给出了使用算法1对表1描述的工作流进行建模的结果。

3　I/O_WF_NET模型化简方法

通过算法1得到的I/O_WF_NET模型中将包含大量的库所和迁移，模型规模较大，不易于后续对其进行分析和利用其生成测试用例，同时，由于工作流中可能出现多个活动由同一输入触发以及多个活动产生相同输出的情况，输入输出关系要进一步理顺，因此需要对其进行化简。

3.1　and-join关系中前驱活动输出相同

如果活动t_i、t_k与活动t_j构成and-join类型的活动关系，且活动t_i、t_k输出内容相同，对工作流中这种结构用算法1进行建模的结果如图3所示，其中$p_{\text{out}_i}=p_{\text{out}_k}=p_{\text{out}}$。在这种情况下，首先将删除迁移$t_i$、$t_k$和库所$p_{\text{out}_i}$、$p_{\text{out}_k}$及其与前后库所、迁移的关系，增加一个新的迁移$t_{ki}$和一个新的库所$p_{\text{out}_{ki}}$，使得${}^{\bullet}t_{ki}=\{p_{\text{in}_k},p_{\text{in}_i}\}$，$t_{ki}^{\bullet}=p_{\text{out}_{ki}}$，$p_{\text{out}_{ki}}^{\bullet}=t_{kj}$，$p_{\text{out}_{ki}}=p_{\text{out}_k}=p_{\text{out}_i}$，如图9所示。

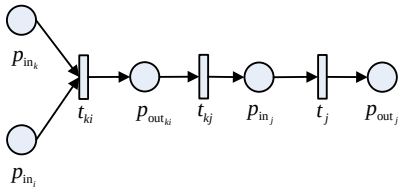


图9　and-join关系化简结果

3.2　or-join关系中前驱活动输出相同

如果活动t_i、t_k与活动t_j构成or-join类型的活动关系，且活动t_i、t_k输出内容相同，对工作流中这种结构用算法1进行建模的结果如图4所示，其中$p_{\text{out}_i}=p_{\text{out}_k}=p_{\text{out}}$。在这种情况下，首先删除虚拟迁移$t_{ij}$、$t_{kj}$和库所$p_{\text{out}_i}$、$p_{\text{out}_k}$及它们与前后库所、迁移的关系，增加一个新的虚拟迁移t_{ikj}和一个新的库所$p_{\text{out}_{ik}}$，使得${}^{\bullet}t_{ikj}=p_{\text{out}_{ik}}$，$t_{ikj}^{\bullet}=p_{\text{in}_j}$，$t_i^{\bullet}=t_k^{\bullet}=p_{\text{out}_{ik}}$，$p_{\text{out}_{ik}}=p_{\text{out}_i}=p_{\text{out}_k}$，如图10所示。

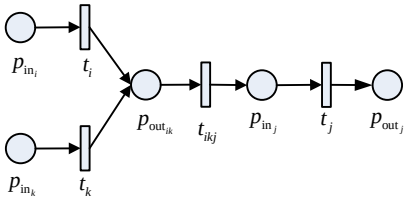


图10　or-join关系化简结果

3.3　and-split关系中后续活动输入相同

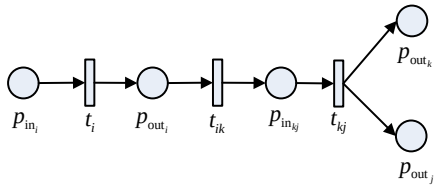


图11　and-split关系化简结果

如果活动 t_i 与活动 t_k、t_j 构成and-split类型的活动关系，且活动 t_k、t_j 由同一输入触发，对工作流中这种结构用算法1进行建模的结果如图5所示，其中 $p_{\text{in}_k} = p_{\text{in}_j} = p_{\text{in}}$。在这种情况下，进行模型简化时，删除库所 p_{in_k}、p_{in_j} 及迁移 t_k、t_j，并删除所有与删除节点有关的关系，增加一个新的库所 $p_{\text{in}_{kj}}$ 和一个新的迁移 t_{kj}，使得 $t_{ik}^{\bullet} = p_{\text{in}_{kj}}$，${}^{\bullet}t_{kj} = p_{\text{in}_{kj}}$，$t_{kj}^{\bullet} = \{p_{\text{out}_k}, p_{\text{out}_j}\}$，$p_{\text{in}_{kj}} = p_{\text{in}_k} = p_{\text{in}_j}$，如图11所示。

3.4　or-split关系中后续活动输入相同

如果活动 t_i 与活动 t_k、t_j 构成or-split类型的活动关系，且活动 t_k、t_j 由同一输入触发，对工作流中这种结构用算法1进行建模的结果如图6所示，其中 $p_{\text{in}_k} = p_{\text{in}_j} = p_{\text{in}}$。在这种情况下，首先删除虚拟迁移 t_{ik}、t_{ij} 和库所 p_{in_k}、p_{in_j} 及它们与前后库所、迁移的关系，增加一个新的虚拟迁移 t_{ikj} 和一个新的库所 $p_{\text{in}_{kj}}$，使得 ${}^{\bullet}t_{ikj} = p_{\text{out}_i}$，$t_{ikj}^{\bullet} = p_{\text{in}_{kj}}$，${}^{\bullet}t_k = {}^{\bullet}t_j = p_{\text{in}_{kj}}$，$p_{\text{in}_{kj}} = p_{\text{in}_k} = p_{\text{in}_j}$，如图12所示。

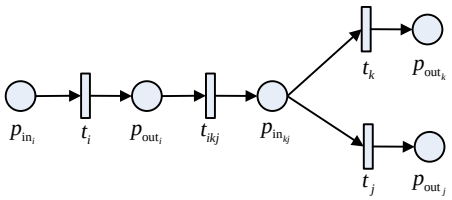


图12　or-split关系化简结果

通过对上述4种情况进行总结不难看出，对“and”类型关系进行化简时，相当于将并行的多个活动合并，对“or”类型关系进行化简时，只是将相同的输入或输出元素合并，而相应活动仍然独立存在。

3.5　虚拟迁移前后库所相同

如果 t_{ij} 是建模后加入到活动 t_i、t_j 之间的虚拟迁移，并且 ${}^{\bullet}t_{ij} = t_{ij}^{\bullet}$，对工作流中这种结构用算法1进行建模的结果如图2所示，其中 $p_{i_{\text{out}}} = p_{j_{\text{in}}}$。在这种情况下，首先删除虚拟迁移 t_{ij} 和库所 $p_{i_{\text{out}}}$、$p_{j_{\text{in}}}$ 及它们与前后库所、迁移的关系；而后增加一个新的库所 p_{ij}，使得 $t_i^{\bullet} = p_{ij}$，${}^{\bullet}t_j = p_{ij}$，$p_{ij} = p_{\text{out}_i} = p_{\text{in}_j}$，如图13所示。这样，库所 p_{ij} 同时表示了活动 t_i 的输出和活动 t_j 的输入，模型包含的信息不发生改变。

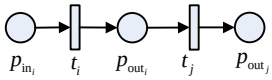


图13　虚拟迁移前后库所相同情况化简结果

表2　对I/O_WF_NET模型化简前后相关信息对比

对比参数	化简前	化简后
活动个数	9	7
库所总个数	20	16
迁移总个数	18	14

通过对I/O_WF_NET模型中上述5类情况的化简得到的新模型不一定就是最简化的，但模型的规模已经小了很多。表2对比了本文例子进行化简前和化简后模型规模的相关参数，图14给出了通过上述方法对图8描述的模型进行简化的结果，可以看出，化简前后模型规模和复杂度均减小了很多。

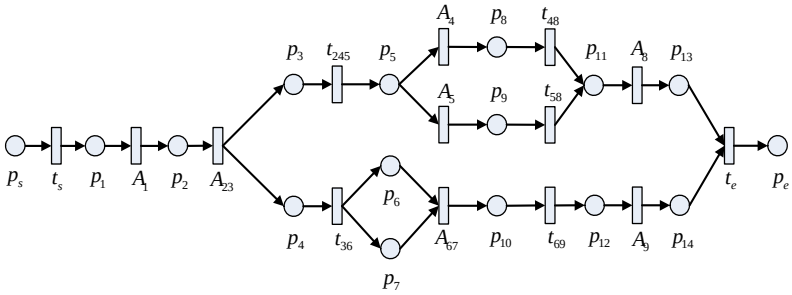


图14　应用化简方法对图8所示模型化简结果

4　总结与展望

近年来，工作流技术快速发展，人们也开始更加关注其功能的正确性和可靠性，这给传统的软件测试带来了新的挑战。本文在给出了具有输入输出约束的工作流网的形式化定义基础上，提出了一种基于Petri网的面向测试的工作流模型I/O_WF_NET，并给出了模型中各组成部分和各种结构的具体建模方法和化简方法，为下一步进行工作流的测试用例自动化生成技术研究打下了基础。未来将主要进行以下3方面的研究：1) 对多输入多输出情况的化简和建模方法进行分析；2) 在本文所建立的

I/O_WF_NET模型的基础上，研究基于该模型的测试用例生成方法；3) 研究基于I/O_WF_NET模型的测试完备性覆盖准则。

参 考 文 献

[1] 周世杰, 秦志光, 刘锦德. 工作流管理系统互操作技术研究[J]. 电子科技大学学报, 2002, 31(2):145-150.
ZHOU Shi-jie, QIN Zhi-guang, LIU Jin-de. Study on interoperability of workflow management system[J]. Journal of University of Electronic Science and Technology of China, 2002, 31(2):145-150.

[2] KARNIEL A, REICH Y. Formalizing a workflow-net implementation of design-structure-matrix-based process planning for new product development[J]. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2011, 47(3): 476-491.

[3] ABRISHAMI S, NAGHIBZADEH M, EPEMA D H J. Cost-driven scheduling of grid workflows using partial critical paths[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 23(8): 1400-1414.

[4] KLAI K, GAALOUL W. Petri net modeling and verification of transactional workflows[C]//20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. Paris, France: IEEE, 2011: 176-184.

[5] TSIRONIS L C, SFIRIS D S, PAPADOPOULOS B K. Fuzzy performance evaluation of workflow stochastic Petri Nets by means of block reduction[J]. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2010, 40(2): 352-362.

[6] HWANG G, LIN C, TSAO L T, et al. A framework and language support for automatic dynamic testing of workflow management systems[C]//Third IEEE International Symposium on Theoretical Aspects of Software Engineering. Tianjin, China: IEEE, 2009: 139-146.

[7] KARAM M, KEIROUZ W, HAGE R. An abstract model for testing mvc and workflow based web applications[C]// Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services. Guadelope, French Southern Territories: IEEE, 2006.

[8] QUAN Lin, LIN Xiao-zhu, WANG Jian-min. An automatic and scalable testing tool for workflow systems[C]//The 3rd International Conference on Grid and Pervasive Computing – Workshops. Kunming, China: IEEE, 2008: 75-80.

[9] BARTZ R. Workflow for automotive test data analysis based on Petri Nets and stored by ASAM ODS[C]//2009 IEEE International Conference on Industrial Technology. Churchill, VIC, Australia: IEEE, 2009: 1-6.

[10] WANG Huai-qing, ZENG Qing-tian. Modeling and analysis for workflow constrained by resources and nondetermined time-an approach based on Petri Nets[J]. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 2008, 38(4): 802-816.

[11] 汤志伟, 殷静. 基于扩展Petri网的仿真建模与分析[J]. 电子科技大学学报, 2012, 41(1): 131-135.
TANG Zhi-wei, YIN Jing, Extended Petri Net-based simulation modeling and analysis[J]. Journal of University of Electronic Science and Technology of China, 2012, 41(1): 131-135.

[12] PETRI C A. Kommunikation mit automaten[D]. Bonn, Germany: Institut fur Instrumentelle Mathematik, 1962.

[13] REISIG W. Petri Nets: an introduction[M]. Berlin, Germany: Springer-Verlag, 1985.

[14] MURATA T. Petri Nets: Properties, analysis and applications[J]. Proceedings of the IEEE, 1989, 77(4): 541-580.

编 辑 漆 蓉