The Language Study for Workflow Net System Based on Petri Net Reduction Technique*

Yan Chun-gang

Jiang Chang-jun

Sun Ping

Li Qi-yan

School of Electronics and Information Engineering, Tongji University, Shanghai, 201804

Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai, 201804

cgyan2@163.com

Abstract: Workflow systems portray the transaction flow process of service process. Workflow models based on Petri net describe the structure of workflow systems intuitively with the form of graphs. Petri net languages reflect the dynamic behavior process of systems. In this paper, it is proved that net reduction rules preserve languages of Workflow net systems based on Petri net reduction technique. Furthermore, a generation algorithm is proposed for the language expression of Workflow net systems. Based on this, a method of performance analysis is given. This method contributes to solve the explosion problem for reachable state space of Petri net to a certain degree, and provides a new algebra technique to analyze Workflow net systems.

Keywords: Petri net, Reduction technique, Workflow net system, Language expression, Performance analysis.

Yan Chun-gang is an associate professor at Tongji University, China. She has presided over or taken part in several research programs, such as Shanghai Science & Technology Research Plan “Grid Architecture and Supporting Technology”. She has published more than 30 papers in the related fields. Her research interests include: concurrent model and algorithm, grid computing model and analysis method.

Jiang Chang-jun is a professor and Ph.D. supervisor at Tongji University. Now, he is also commissioner of Science & Technology of Shanghai, deputy director of the Special Interest Committee of Petri net, and director of School of Electronics & Information Engineering, Tongji University. Currently, his research interests include grid computing research, parallel computing, and Petri net theory.

Sun Ping is a Ph.D. candidate at Tongji University. Her research interests include Petri net, Semantic Grid and workflow theory.

Li Qi-yan is a professor and Ph.D. supervisor at Tongji University, and the director of the research center of CAD technology. His research interests include graph design, computed aided design, and engineering data management.

1. Introduction

Workflow model is the basis of Workflow Management Systems, there has been an increased interest in the formal modeling of workflow processes, such as flow chart, Pi-Calculus, UML, and Petri net. With the advantages of formal semantics definition, graphical nature, firm mathematical foundation and analysis techniques, Petri net is frequently used to model and analyze workflow process.

Aalst[1] builds a formal model to describe workflow process, namely workflow net, and defines soundness of workflow net. He further discusses the structural property and analysis methods for workflow net. [2]

* Supported by the National Natural Science Foundation of China under Grant No.60473094, the National Basic Research Program of China under Grant No.2003CB316902.

proposes a set of graph reduction rules to verify the soundness of free choice workflow nets. [3] defines reduction rules accomplishing semantic verification based on three dimensional workflow net.

Workflow net is a special kind of Petri net, so the analysis method of Petri net can also be applicable to workflow net system. Reduction technique is one of the effective means for Petri net analysis, for example, [4] studies the reduction rules of T-graph, [5] applies Petri net reduction to Ada Task Deadlock analysis. However, most of previous works on Petri net reduction are restricted to structural preservation. In fact, language expression is also another valid method, and there are many research on this field[6-13]. This method is helpful to analyze the dynamic behavior and performance without the reachable state graph. Therefore, it is important to generate the language expression of Petri net. Unfortunately, it is difficult for some complex nets, and the state explosion problem forbids its application.

Most of previous research of workflow net focus on the modeling and structural analysis, and there is no work on the language of workflow net system. As the workflow net is a special kind of Petri net, it will be benefic if we take the reduction technique into account for workflow net system analysis. So we aim at the reduction rules, to discuss the problem of workflow language generation. Firstly, we give six reduction rules, and prove that these rules preserve the language of workflow net system. Secondly, we propose a generation algorithm of language expression. Lastly, we analyze the performance of workflow process. A case study is given to illustrate our method. This method contributes to solve the explosion problem for reachable state space of Petri net to a certain degree, and provides a new algebra technique to analyze Workflow net systems.

2. Preliminaries

Definition 1 [14-15]. Let *T* be a finite set of symbols, $T*$ be the set of finite sequence of symbols, then for $\forall L \subseteq T*$, *L* is called the language over *T*.

Definition 2 [14-15]. Let *T* be a finite symbols set, L_1, L_2 be the language over *T*, we define the following operators:

(1) Sequence Operator "$\circ$": $L_1 \circ L_2 = \{\alpha_1 \circ \alpha_2 \big| \alpha_1 \in L_1 \wedge \alpha_2 \in L_2\}$;

(2) Choice Operator "+": $L_1 + L_2 = \{\alpha \big| \alpha \in L_1 \vee \alpha \in L_2\}$;

(3) Parallel Operator "$//$": $L_1 // L_2 = \{\alpha_1 // \alpha_2 \big| \alpha_1 \in L_1 \wedge \alpha_2 \in L_2\}$, for $a, b \in T$, $\alpha_1, \alpha_2 \in T*$ and ε, $a\alpha_1 // b\alpha_2 = a(\alpha_1 // b\alpha_2) + b(a\alpha_1 // \alpha_2)$ and $a // \varepsilon = \varepsilon // a = a$;

(4) Loop Operator "*": $L_1^* = \bigcup_{x=1}^{\infty} L_1^x$, where $L_1^0 = \{\varepsilon\}$, $L_1^x = L_1 \circ L_1^{x-1}$, $x \geq 1$.

Definition 3 [14-15]. Let *T* be a finite symbols set, the language over *T* and the sets they denote are defined recursively as follows:

(1) ϕ is a language expression and denotes the empty set;

(2) ε is a language expression and denotes the set $\{\varepsilon\}$;

(3) For $\forall a \in T$, a is a language expression and denotes the set $\{a\}$;

(4) If α_1, α_2 are the language expressions denoting the language L_1, L_2, then $\alpha_1 \circ \alpha_2$, $\alpha_1 + \alpha_2$, $\alpha_1 // \alpha_2$ and α_1^* are the language expression that denote the sets $L_1 \circ L_2$, $L_1 + L_2$, $L_1 // L_2$ and L_1^* respectively.

In this paper, we denote a Petri net as $\Sigma = (P, T; F, M_0)$, for basic introduction to Petri net, please refer to [13,14].

Definition 4 [1]. Let Petri net $\Sigma = (P, T; F, i)$, if

(1) There is a source place $i \in P$, i.e., ${}^{\bullet}i = \phi$;

(2) There is a sink place $o \in P$, i.e., $o^{\bullet} = \phi$;

(3) For every node $x \in P \cup T$, there is a directed path from i to o via x;

(4) The initial state is that there is only one token in place i except other places.

Then $\Sigma = (P, T; F, i)$ is called a workflow net system.

The reachable state set of Σ is denoted as $RSS(\Sigma)$, where any reachable state is described as a bag[13], for example $s \in RSS(\Sigma) : s = \{p_3, 2p_5, p_6\}$.

3. Reduction Technique of Workflow Net Language

In this section, we will discuss the reduction technique of workflow net language. At first, we will give six reduction rules, including sequence reduction, choice reduction and so on. Then we will prove that these rules preserve the language of workflow net system.

Definition 5 (Sequence Reduction). Let $\Sigma = (P, T; F, i)$ be a workflow net system, if $\exists a, b \in T \rightarrow a^{\bullet} = {}^{\bullet}b$, set the reduced net system $\Sigma' = (P', T'; F', i')$, such that:

$P' = P - {}^{\bullet}b$,

$T' = T \cup \{ab\} - \{a, b\}$,

$F' = F \cup \{(p, ab), (ab, q) \mid \forall p \in {}^{\bullet}a, \forall q \in b^{\bullet}\} - \{(p, a), (b, q) \mid \forall p \in {}^{\bullet}a, \forall q \in b^{\bullet}\}$,

$i' = i$.

Then it is called the sequence reduction, the reduction rule is denoted as R1, which is shown in Figure1.

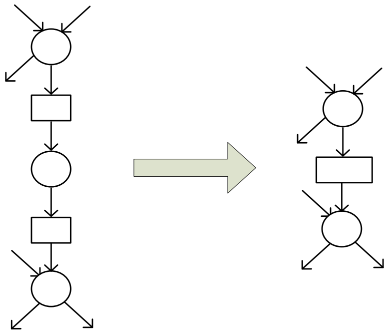


Figure 1. Sequence Reduction

a

R1

Definition 6 (Choice Reduction). Let $\Sigma = (P,T;F,i)$ be a workflow net system, if $\exists a,b \in T \to {}^\bullet a = {}^\bullet b \neq \phi \wedge a^\bullet = b^\bullet \neq \phi$, set the reduced net system $\Sigma' = (P',T';F',i')$, such that:

$P' = P$,

b

$T' = T \cup \{a+b\} - \{a,b\}$,

$F' = F \cup \{(p,a+b),(a+b,q) \mid \forall p \in {}^\bullet a \cup {}^\bullet b, \forall q \in a^\bullet \cup b^\bullet\}$

$- \{(p,a),(p,b),(a,q),(b,q) \mid \forall p \in {}^\bullet a \cup {}^\bullet b, \forall q \in a^\bullet \cup b^\bullet\}$,

$i' = i$.

Then it is called the choice reduction, the reduction rule is denoted as R2, which is shown in Figure2.

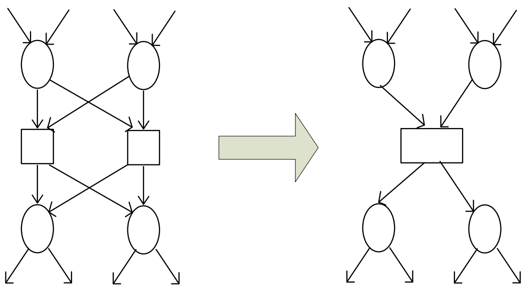


Figure 2. Choice Reduction

Definition 7 (Parallel Reduction). Let $\Sigma = (P,T;F,i)$ be a workflow net system, if $\exists a,b \in T \to {}^\bullet a \cap {}^\bullet b = \phi \wedge a^\bullet \cap b^\bullet = \phi$, set the reduced net system $\Sigma' = (P',T';F',i')$, such that:

$P' = P$,

$T' = T \cup \{a // b\} - \{a,b\}$,

$F' = F \cup \{(p,a // b),(a // b,q) \mid \forall p \in {}^\bullet a \cup {}^\bullet b, \forall q \in a^\bullet \cup b^\bullet\}$

$-\{(p',a),(p'',b),(a,q'),(b,q'')\,|\,\forall p' \in {}^{\bullet}a, \forall p'' \in {}^{\bullet}b, \forall q' \in a^{\bullet}, q'' \in b^{\bullet}\}$,

$i' = i$.

Then it is called the parallel reduction, the reduction rule is denoted as R3, which is shown in Figure3.

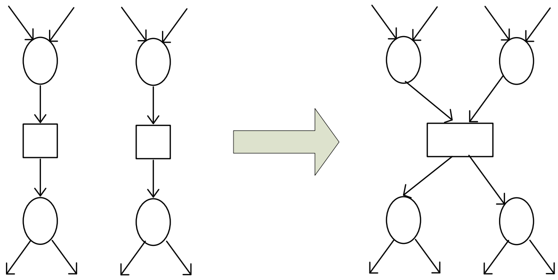


Figure 3. Parallel Reduction

Definition 8 (Loop Reduction). Let $\Sigma = (P,T;F,i)$ be a workflow net system, if $\exists a,b \in T \rightarrow$ $|{}^{\bullet}a| = |{}^{\bullet}b| = |a^{\bullet}| = |b^{\bullet}| = 1 \wedge {}^{\bullet}a = b^{\bullet} \wedge a^{\bullet} = {}^{\bullet}b$, set the reduced net system $\Sigma' = (P',T';F',i')$, such that:

$P' = P$,

$T' = T \cup \{(ab)*a\} - \{a,b\}$,

$F' = F \cup \{(p,(ab)*a),((ab)*a,q)\,|\,\forall p \in {}^{\bullet}a, \forall q \in a^{\bullet}\} - \{(p,a),(a,q),(q,b),(b,p)\,|\,\forall p \in {}^{\bullet}a, q \in a^{\bullet}\}$, R

a b

$i' = i$.

Then it is called the loop reduction, the reduction rule is denoted as R4, which is shown in Figure4.

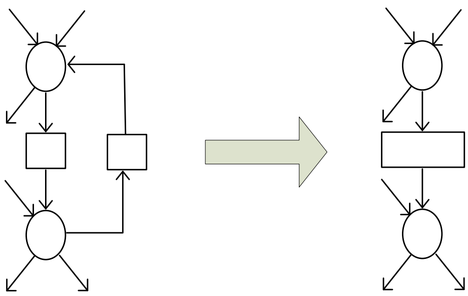


Figure 4. Loop Reduction

Definition 9 (Circle Reduction). Let $\Sigma = (P,T;F,i)$ be a workflow net system, if $\exists a,b \in T \rightarrow {}^{\bullet}a = a^{\bullet} = {}^{\bullet}b \neq b^{\bullet} \wedge |{}^{\bullet}a| = |a^{\bullet}| = |{}^{\bullet}b| = |b^{\bullet}| = 1$, set the reduced net system $\Sigma' = (P',T';F',i')$, such that:

$P' = P$,

$T' = T \cup \{(a)*b\} - \{a,b\}$,

$F' = F \cup \{(p,(a)*b),((a)*b,q) \big| \forall p \in {}^\bullet b, \forall q \in b^\bullet\} - \{(p,a),(a,p),(p,b),(b,q) \big| \forall p \in {}^\bullet b, q \in b^\bullet\}$,

$i' = i$.

Then it is called the circle reduction, the reduction rule is denoted as R5, which is shown in Figure5.

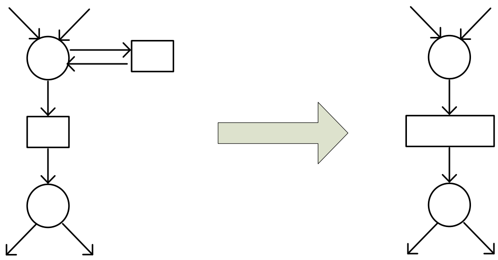


Figure 5. Circle Reduction

Definition 10 (Redundant Reduction). Let $\Sigma = (P,T;F,i)$ be a workflow net system, $a \in T$, if

(1) $\forall p_i, p_j \in {}^\bullet a \rightarrow {}^\bullet p_i = {}^\bullet p_j \wedge p_i^\bullet = p_j^\bullet$, then set $\{p_i\} = {}^\bullet a$, and others remain unchanged to get the reduced net system $\Sigma' = (P',T';F',i')$;

a

or (2) $\forall p_i, p_j \in a^\bullet \rightarrow {}^\bullet p_i = {}^\bullet p_j \wedge p_i^\bullet = p_j^\bullet$, then set $\{p_i\} = a^\bullet$, and others remain unchanged to get the reduced net system $\Sigma' = (P',T';F',i')$.

b

Then it is called the redundant reduction, the reduction rule is denoted as R6, which is shown in Figure6.

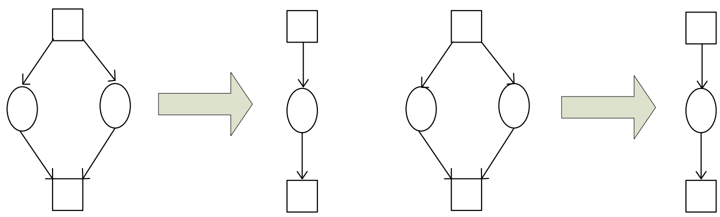


Figure 6. Redundant Reduction

3.2 Language Preservation Property of Reduction Rules

Petri net language is a powerful tool for describing dynamic behavior of physical systems. However, it is not easy to obtain the language expression for a given Petri net especially a structure-complex net. In this paper, we will propose an algorithm to generate the language expression based on reduction technique, to solve the problem of state explosion. In order to guarantee the correctness of this algorithm, it is necessary that the reduction rules should preserve the language of net system. In the following, we will prove this conclusion.

Definition 11. Let $\Sigma = (P,T;F,i)$ be a workflow net system, $RSS(\Sigma)$ be the reachable state set of Σ . Set $L(\Sigma) = \{\sigma \,|\, \sigma \in T^* \wedge i[\sigma > M \wedge \forall M \in RSS(\Sigma)\}$, then $L(\Sigma)$ is called the language of Σ .

Theorem 1. Let $\Sigma = (P,T;F,i)$ be a workflow net system, R ∈ {R1, R2, R3, R4, R5, R6},

$\Sigma' = (P', T'; F', i')$ be the reduced net system by rules R, then $L(\Sigma) = L(\Sigma')$.

Proof: For $\forall \sigma \in L(\Sigma)$, if $\{a,b\} \cap \|\sigma\| = \phi$ (where $\|\sigma\|$ returns the set of symbols in σ), then R has no relationship with σ, so $\sigma \in L(\Sigma')$. Otherwise, $\exists i, s_1, s_2, s_3, s_4 \in RSS(\Sigma)$, such that:

(1) For rule R1, $i[\sigma > s_4$, i.e., $i[\sigma_1 > s_1[a > s_2[b > s_3[\sigma_2 > s_4$. According to Defintion5, $i' = i, s_1' = s_1, s_3' = s_3, s_4' = s_4$, satisfying $i'[\sigma_1 > s_1'[ab > s_3'[\sigma_2 > s_4'$, that is to say $i'[\sigma > s_4'$, where $i', s_1', s_2', s_3', s_4' \in RSS(\Sigma')$, therefore $\sigma \in L(\Sigma')$.

(2) For rule R2, $i[\sigma > s_3$, i.e., $i[\sigma_1 > s_1[a > s_2[\sigma_2 > s_3$ or $i[\sigma_1 > s_1[b > s_2[\sigma_2 > s_3$. According to Defintion6, $i' = i, s_1' = s_1, s_2' = s_2, s_3' = s_3$, satisfying $i'[\sigma_1 > s_1'[a + b > s_2'[\sigma_2 > s_3'$, that is to say $i'[\sigma > s_3'$, where $i', s_1', s_2', s_3' \in RSS(\Sigma')$, therefore $\sigma \in L(\Sigma')$.

(3) For rule R3, $i[\sigma > s_3$, i.e., $i[\sigma_1 > s_1, s_1[a >$ and $s_1[b >$, $s_1[ab > s_2[\sigma_2 > s_3$ or $s_1[ba > s_2[\sigma_2 > s_3$. According to Defintion7, $i' = i, s_1' = s_1, s_2' = s_2, s_3' = s_3$, satisfying $i'[\sigma_1 > s_1'[a // b > s_2'[\sigma_2 > s_3'$, that is to say $i'[\sigma > s_3'$, where $i', s_1', s_2', s_3' \in RSS(\Sigma')$, therefore $\sigma \in L(\Sigma')$.

(4) For rule R4, $i[\sigma > s_3$, i.e., $i[\sigma_1 > s_1[a > s_2[b > s_1[a > s_2[... > s_1[a > s_2[b > s_1[a > \quad s_2[\sigma_2 > s_3$. According to Defintion8, $i' = i, s_1' = s_1, s_2' = s_2, s_3' = s_3$, satisfying $i'[\sigma_1 > s_1'[(ab)^* a > s_2'[\sigma_2 > s_3'$, that is to say $i'[\sigma > s_3'$, where $i', s_1', s_2', s_3' \in RSS(\Sigma')$, therefore $\sigma \in L(\Sigma')$.

(5) For rule R5, it can be proved by (2) and (4).

(6) For rule R6, $i[\sigma > s_3$, i.e., $i[\sigma_1 > s_1[a > s_2[\sigma_2 > s_3$. According to Defintion10, $i' = i$,

$s_1' = s_1 \cup \{p_i\} - \{p_j \mid (\forall p_j \in {}^\bullet a) \wedge ({}^\bullet p_j = {}^\bullet p_i \wedge p_j^\bullet = p_i^\bullet \wedge p_i \in a^\bullet, a \in T'(=T)\}$,

$s_2' = s_2 \cup \{p_i\} - \{p_j \mid (\forall p_j \in a^\bullet) \wedge ({}^\bullet p_j = {}^\bullet p_i \wedge p_j^\bullet = p_i^\bullet \wedge p_i \in a^\bullet, a \in T'(=T)\}$, $s_3' = s_3$, satisfying

$i'[\sigma_1 > s_1'[a > s_2'[\sigma_2 > s_3'$, that is to say $i'[\sigma > s_3'$, where $i', s_1', s_2', s_3' \in RSS(\Sigma')$, therefore $\sigma \in L(\Sigma')$.

In a summary, we can draw the conclusion that $L(\Sigma) \subseteq L(\Sigma')$, similarly $L(\Sigma') \subseteq L(\Sigma)$. As a consequence, $L(\Sigma) = L(\Sigma')$.

4. Language Expression of Workflow Net System Generation Algorithm

For a simply workflow net, the language expression can be obtained by means of reduction, however, it is difficult and impossible for some complex nets.

Theorem 2. Let $\Sigma = (P, T; F, i)$ be a workflow net system, if Σ is bounded, then it has language expression.

Proof: If Σ is bounded, then its reachable state graph $RMG(\Sigma)$ is equivalent to a finite state automaton, therefore there is a normal expression equivalent to the language of $RMG(\Sigma)$, which is the language expression of workflow net system.

It is well known that it is difficult to seek the reachable state graph for a workflow net. If we can simply the workflow net by reduction rules at first, to get an equivalent net system, and calculate the reachable graph, then we will generate the language expression of the original net system much more efficiently. The following is the algorithm to generate the language expression for the workflow net system.

Algorithm 1. Generation Algorithm of Language Expression for Workflow Net System

Input: $\Sigma = (P, T; F, i)$

Output: $\alpha(\Sigma)$

Step1. Reduce the workflow net system Σ by rules R1-R6 step by step, the reduced net system is R(Σ);

Step2. If R(Σ) is the most simple workflow net system, then the algorithm ends, the language expression of Σ is $\alpha(R(\Sigma))$; Otherwise, go to Step3;

Step3. Generate reachable state graph $RMG(\mathrm{R}(\Sigma))$ of R(Σ);

Step4. Simply $RMG(\mathrm{R}(\Sigma))$ by means of the reduction rules of finite state automation, to generate the language expression $\alpha(\Sigma)$ of workflow net system Σ .

Theorem 3. Let $\Sigma = (P, T; F, i)$ be a workflow net system, $\alpha(\Sigma)$ be the language expression generated by algorithm1, then $L(\Sigma) = L(\alpha(\Sigma))$.

Proof: The conclusion can be easily proved by Theorem 1 and the theory of finite state automaton [13].

Example 1. A workflow net system $\Sigma_1 = (P_1, T_1; F_1, i_1)$ is shown in Figure 7, the language expression generation process is given in Figure8-Figure10 according to Algorithm 1.

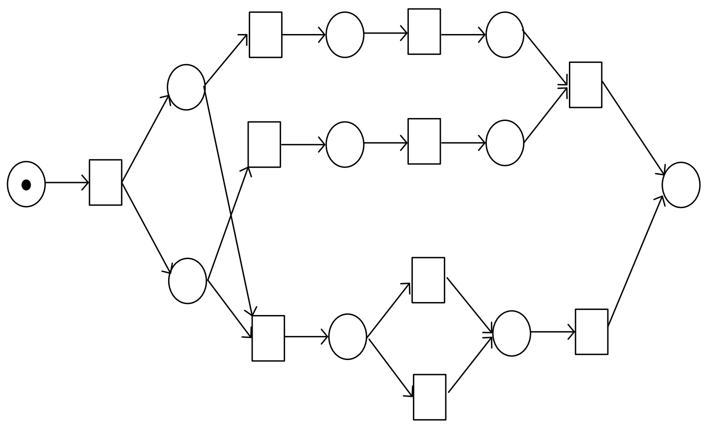
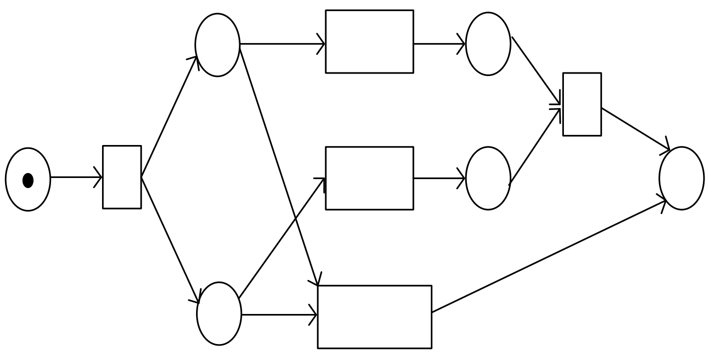


Figure 7. Workflow Net System Σ_1



b

Figure 8. Reduced Workflow Net System R(Σ_1 **)**

p4

p2

c

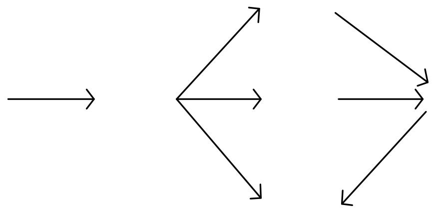


Figure 9. Reachable State Graph $RMG(\mathrm{R}(\Sigma_1))$ **of R(**Σ_1**)**

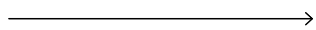


Figure 10. Reduced $RMG(\mathrm{R}(\Sigma_1))$

It is clear that the language expression of this workflow net system Σ_1 is $\alpha(\Sigma_1) = a((be)//(cf)i + d(g+h)j)$.

5. Performance Analysis based on Workflow Language

5.1 Performance Analysis

Performance analysis is one of the most important aspects in the research of workflow technology. It is to evaluate the ability to meet requirements with respect to throughput times, resource utilization rate, and so on. For a practical workflow net, the problem of state explosion often hinders the performance analysis. Here, we put forward a performance analysis method for workflow net system. It makes use of the Stochastic Petri net analysis method based on language expression [14]. This approach can not only apply to arbitrary distribution firing delay, but also decrease the analysis complexity.

The detailed steps include:

Step 1: According to Algorithm1, construct the language expression of the system, and transform it into the form of standard polynomial;

Step 2: Calculate the firing probability, moment function and transfer function of each transition step by step; and remark the expression to distinguish the same transition with different transfer function;

Step 3: Obtain the transfer function of the whole expression;

Step 4: Evaluate the performance parameters of the system.

5.2 Case study

Example 2 [1]. Figure 11 shows a workflow net system $\Sigma_2 = (P_2, T_2; F_2, i_2)$ for the processing of complaints. First the complaint is registered (task register), then in parallel a questionnaire is sent to the complainant (task send-questionnaire) and the complaint is evaluated (task evaluate). If the complaint returns the questionnaire within two weeks, the task process-questionnaire is executed. If the questionnaire is not returned within two weeks, the result of the questionnaire is discarded (task time-out). Based on the result of the evaluation, the complaint proceeds or not. The actual processing of the complaint (task process-complaint) is delayed until the questionnaire is processed or a time-out has occurred. The processing of the complaint is checked via task-processing. Finally, task archive is executed.

The explanation and firing delay for every transition is shown in Table 1.

Table 1. Explanation and Firing Delay for every Transition

a: register	[a=1,b=2] uniform distribution
b: send-questionnaire	$\lambda_b = 4$ exponential distribution
c: evaluate	$\lambda_c = 0.5$ exponential distribution
d: time-out	immediate transition, firing probability q_1=0.2
e: process-questionnaire	$\lambda_e = 0.5$ exponential distribution
f: no-processing	immediate transition, firing probability q_2=0.3
g: archive	$[\mu_1 = 0.5, \sigma_1 = 0.4]$ normal distribution
h: processing-required	immediate transition, firing probability 1-q_2
i: process-complaint	$\lambda_i = 0.1$ exponential distribution
j: check- processing	$\lambda_j = 0.1$ exponential distribution
k: processing-NOK	immediate transition, firing probability q_3=0.2
l: processing-OK	immediate transition, firing probability 1-q_3

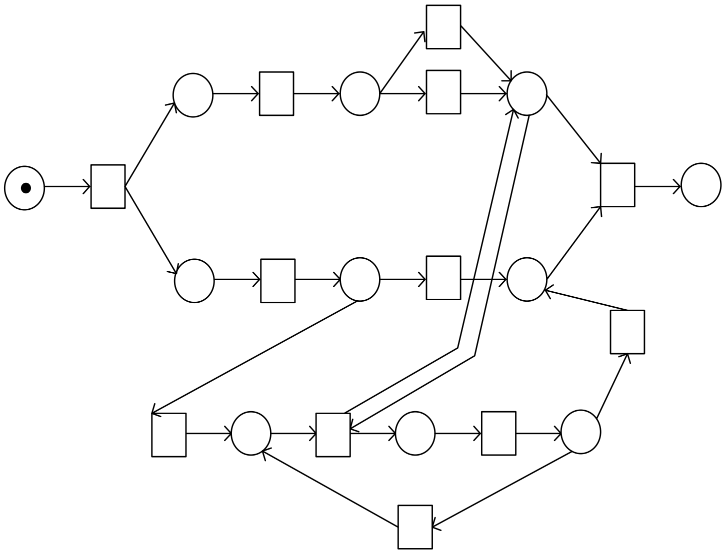
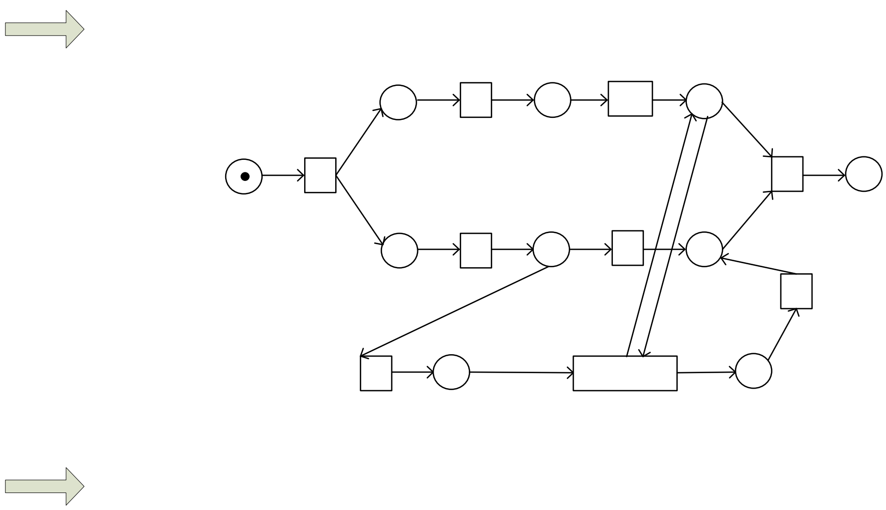
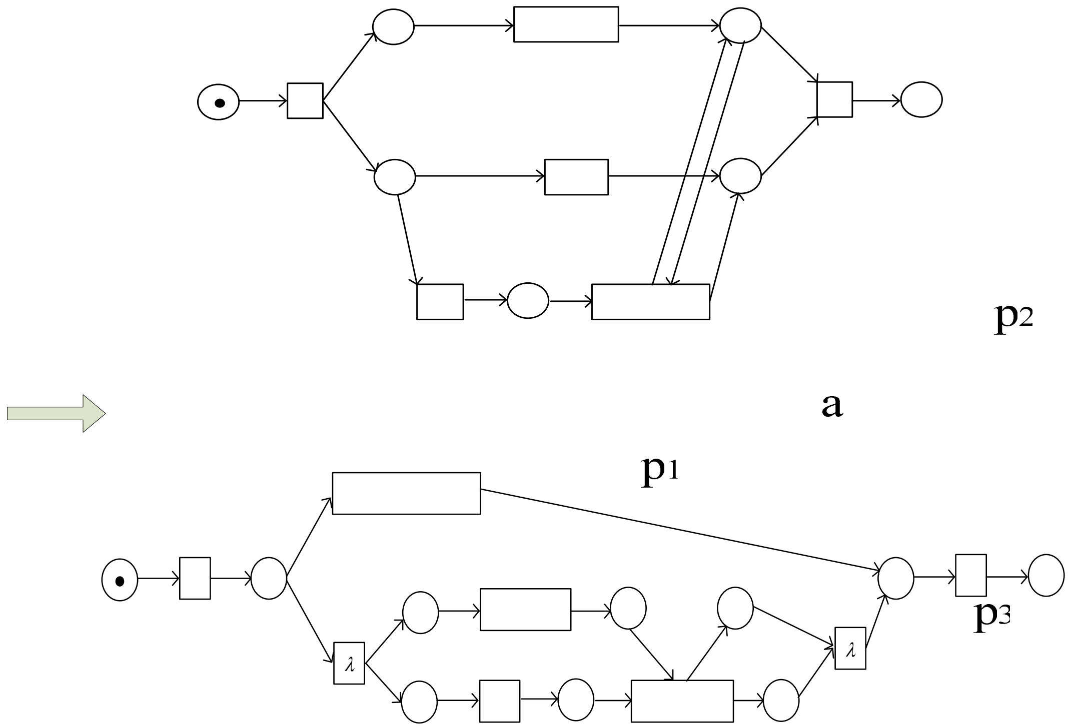


Figure 11. Workflow Net System Σ_2 for the Processing of Complaints

As this workflow net system is a little complex, and the firing delays of transitions don't obey exponential distribution, so we utilize the Stochastic Petri net analysis method based on language expression to evaluate the performance indices.

(1) Firstly, we simply the workflow net system with the help of the reduction rules in Section 3, in order to obtain the language expression. The rough reduction process is shown in Figure 12:





Here, we add two transitions λ to denote "OR-Split" and "OR-Join", which don't stand for any operation.
Meanwhile, we split place p_6 as p_6 as p_6' , in order to eliminate self-circle.

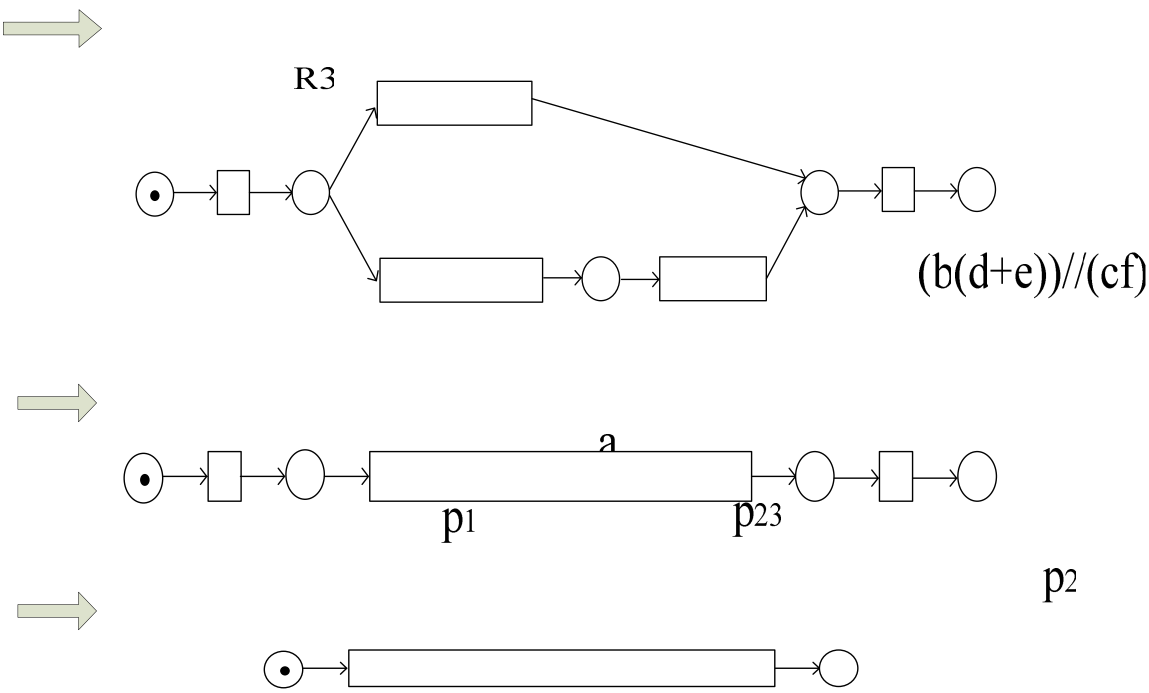


Figure 12. Reduction Process for Workflow net System Σ_2

So the language expression of Σ_2 is $\alpha(\Sigma_2) = a((b(d+e))//(cf) + (b(d+e))//(ch)(ijk)^* ijl)g$. Then we remove the immediate transition, simply $\alpha(\Sigma_2)$ as $a((be)//c + ((be)//c)(ij)^* ij)g$, and transform it to standard polynomial, that is : $\alpha(\Sigma_2) = a(bec + bce + cbe + (bec + bce + cbe)(ij)^* ij)g$.

(2) Remark $\alpha(\Sigma_2) = a(b^1e^1c^1 + b^1c^2e^2 + c^3b^2e^2 + (b^3e^3c^4 + b^3c^5e^4 + c^6b^4e^4)(ij)^* ij)g$, where the transfer functions $W_a(s) = \int_a^b xe^{sx}dx = \frac{e^{bs} - e^{as}}{(b-a)s}$, $W_{b1}(s) = q_2 \frac{\lambda_b}{\lambda_b + \lambda_c - s}$, and so on.

(3) According to the equivalent transfer function of the language expression $W_\alpha(s)$, the mean execution time of workflow process can be obtained as $T = \frac{\partial}{\partial s} W_\alpha(s) |_{s=0} \approx 15.28$ (hours). Based on this, we can calculate other performance indices, such as throughput, resource utilization rate, and so on.

6. Conclusion

Focusing on the language of workflow net system in order to analyze the property is an important approach, while there is no work on this field. In this paper, it is proved that net reduction rules preserve languages of workflow net systems. Furthermore, a generation algorithm is proposed for the language expression of workflow net systems. This method contributes to solve the explosion problem for reachable state space to a certain degree. What's more, this result can also provide firm support for the analysis of dynamic behavior and performance evaluation of workflow net system. The work presented here is a starting point and there are numerous possible extensions. For example, we intend to study other useful means to generate language expression, and other effective methods to analyze workflow net system.

REFERENCES

1. VAN DE ALST, W.M.P, **Workflow Management—Models, Methods and Systems**, Beijing: Tsinghua Press, 2004.
2. LI JIANQIANG, FAN YUSHUN, **Research of Petri net based Workflow Model Reduction Methods**, Information and Control, Vol. 30, No. 6, 2001.
3. ZHOU JIANTAO, SHI MEILIN, YE XINMING, **A Method for Semantic Verification of Workflow Processes Based on Petri Net Reduction Technique**, Journal of Software, Vol. 16, No. 7, 2005.
4. XU ANGUO, JIANG CHANGJUN, **The Reduction Operations and their Properties for P/T nets**, Journal of Software, Vol. 8, No. 7, 1997.
5. SOL M.SHATZ, TADAO MURATA, **An Application of Petri Net Reduction for Ada Tasking Deadlock Analysis**, IEEE Transactions on Parallel and Distributed System, Vol. 7 No. 12, 1996.

6. LI JIAO, TO-YAT CHEUNG, WEIMING LU, **Handing Synchronization Problem in Petri Net-Based System Design by Property-Preserving Transition Reduction**, The Computer Journal, 2005, Vol. 48, No. 6, 2005.

7. JIANG, C. J., **A Polynomial-time Algorithm for the Legal Firing Sequences Problem of A Type of Synchronous Composition Petri Nets**, Science in China (Series E), Vol. 44, No. 3, 2001.

8. JIANG, C. J., **Testing of Functions of Complex Systems Based on Synchronous Composition Nets**, International Journal of Studied on Information Control, Vol. 9, No. 4, 2000.

9. JIANG, C.J., et. al., **Behaviour Relations in Synthesis Process of Petri Net Models**, IEEE Trans. on RA, Vol. 16, No. 4, 2000.

10. JIANG, C. J., et. al., **Urban Traffic Information Service Application Grid**, Journal of Computer Science & Technology, Vol. 20, No. 1, 2005.

11. JIANG, C. J.,.WANG, H. Q,. LIAO, S. Y, **Behavior Relativity of Petri Nets**, Journal of Computer Science & Technology, Vol. 17, No. 6, 2002.

12. DU, Y. Y., JIANG, C. J., **Formal Representation and Analysis of Batch Stocks Trading Systems through Logical Petri Net Workflows**, Proceedings of 4th International Conference on Formal Engineering Methods, Lecture Notes in Computer Science, No. 2495, 2002.

13. PETERSON, J. L. (Wu Zhehui translation), **Petri net Theory and the Modeling of Systems**, Xuzhou: China University of Mining and Technology Press, 1989.

14. JIANG CHANGJUN, **A PN Machine Theory of Discrete Event Dynamic System**, Beijing:Science Press, 2000.

15. HOPCROFT, J. E. (Xu Meirui translation), **Introduction to Automate Theory, Language and Computation**, Beijing: Science Press, 1990.