Soundness Analytics of Composed Logical Workflow Nets

Wei Liu[1,2] · **Lu Wang**[2] · **Xin Feng**[2] · **Man Qi**[3] · **Chun Yan**[4] · **Maozhen Li**[5,6]

Abstract Cooperative systems with passing value indeterminacy and batch processing can be well modeled by composed logical workflow nets. Soundness guarantees no deadlock and livelock and each activity has potential to be executed. The soundness of composed logical workflow nets can be judged by reachability graphs. But reachabil-ity graphs can cause state space explosion. Path nets, single line nets, composed path nets and composed single line nets are proposed in the paper. They are used to deter-mine soundness of logical workflow nets and composed logical workflow nets based on net structures and logical expressions avoiding reachability graphs. The presented concepts and techniques are applied to judge soundness of e-commerce transaction processes modeled by composed logical workflow nets, and they are illustrated by an example.

B Chun Yan
yanchunchun9896@163.com

1 Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China

2 College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China

3 Computing, Digital Forensics and Cybersecurity, Canterbury Christ Church University, Canterbury, Kent CT1 1QU, UK

4 College of Mathematics and System Science, Shandong University of Science and Technology, Qingdao, China

5 Department of Electronic and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, UK

6 Key Laboratory of Embedded Systems and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China

Keywords Net structure · Soundness · Composed logical workflow nets · Logical expression

1 Introduction

Workflow nets are a subclass of Petri nets [1]. They are widely used in modeling and analysis of business systems [2]. With development of Internet, communications between organizations are convenient and each organization is no longer isolated. Many business processes are across organizations. Multiple organizations often coop-erate each other[3,4]. Thus, modeling and analysis of business processes across organizations are crucial [5]. Composed logical workflow nets can well describe the features of systems, especially synchronous and asynchronous interactions among multiple systems.

The importance of modeling methods and techniques of composed logical workflow nets are discussed in many research literatures [6]. Ordinary composed workflow nets cannot well show passing value indeterminacy among multiple organizations and batch processing of systems [7]. Composed logical workflow nets can well describe the two features of system processes by adding logical transitions [8]. The firing of some logical transitions of composed logical workflow nets does not necessarily need all resources of preset places [9]. As long as the logical expression corresponding to a logical transition is satisfied, the partial resources in preset places can also be processed [10]. This speciality can be applied in modeling composed Web services [11–14], e-commerce transaction processes and son on [15]. It reveals even if not all data of a task arrive before a deadline, the task can be executed to improve efficiency. Later arrival data can be processed in the business process next time.

For an organization, the correctness of its business process is vitally important [16,17]. Soundness is an important property of composed logical workflow nets used to determine the correctness of business processes. Soundness of composed logical workflow nets means no dead transition, no deadlock, no livelock in the model and correct end states can be reached. The reachability graph can be used to determine soundness of logical workflow nets. The reachability graph is intuitive and easy to be implemented. However, with the increase in size of composed logical workflow nets, the complexity grows exponentially. If net structures [18,19] and logical expressions are applied to determine soundness of logical workflow nets, the complexity can be reduced.

In this paper, path nets and single line nets are proposed based on net structures and logical expressions of logical workflow nets. Composed path nets and composed single line nets are presented according to net structures of composed logical workflow nets. Composed path nets and composed single line nets are obtained based on path nets and single line nets of each logical workflow nets. Soundness of logical workflow nets can be judged by determining if single line nets can be included in path nets and logical workflow nets can be covered by path nets. Soundness of composed logical workflow nets can be induced by determining the relations between composed path nets and composed single line nets. These methods are based on net structures and logical expressions, and do not generate reachability graphs of logical workflow nets and composed logical workflow nets.

The rest is organized as follows. Section 2 briefly introduces logical workflow nets and their soundness. Composed logical workflow nets and their soundness are presented in Sect. 3. Section 4 proposes path nets and single line nets of logical workflow nets and the method of determining soundness of logical workflow nets is put forward. Section 5 proposes composed path nets and composed single line nets of composed logical workflow nets. The method of determining soundness of composed logical workflow nets is presented. Section 6 models a business workflow in electronic commerces. The presented concepts and methods are illustrated with this example and soundness is judged by the proposed techniques. Section 7 concludes this paper.

2 Logical Workflow Nets

The related concepts of logical workflow nets are introduced in this section.

2.1 The Definition of Logical Workflow Nets

Definition 1 A logical workflow net is a six-tuple LWF_net = $(P, T, I, O, F; M_0)$, where

(1) $P = P_t \cup P_d$, P_t is a finite control place set, P_d is a finite data place set; i and o are two special places, i denotes the source place, ${}^{\bullet}i = \varphi$; o is the end place, $o^{\bullet} = \varphi$.

(2) $T = T_G \cup T_O \cup T_I$, T_G is an ordinary transition set; T_O is a logical output transition set,and each logical output transition is subject to a logical output expression f_O; T_I denotes a logical input transition set,and each logical input transition is subject to a logical input expression f_I.

(3) $F \subseteq (P \times T) \cup (T \times P)$, F denotes a directed arc set between places and transitions.

(4) I is a logical input function mapping a logical input transition to a logical input expression, i.e., $\forall t \in T_I$, $I(t) = f_I(t)$;

(5) O is a logical output function from a logical output transition to a logical output expression, i.e., $\forall t \in T_O$, $O(t) = f_O(t)$;

(6) M denotes a state, $M(p)$ denotes the number of tokens in a place p; M_0 denotes a initial state, $M_0 = i$ denotes only the source place includes a token and other places include no token; M_D denotes a end state, $M_D = o$ denotes only the end place includes a token and other places include no token;

(7) Below are two kinds of logical relation in logical expressions:

:Union operator, Let $f_I = a \wedge b$, if both a and b include tokens, $f_I|_M = T$; if a or b includes no token, $f_O|_M = F$;Let $f_O = a \wedge b$, if both a and b include tokens, $f_O|_M = T$, if a or b includes no token, $f_O|_M = F$.

:substitution operator, Let $f_I = a\ b$, if both a and b include tokens, $a\ b = a \wedge b$, $f_I|_M = T$;if a include a token, b include no token, $a\ b = 1 \vee 0$, $f_I|_M = T$;if a includes no token, b includes a token, $a\ b = 0 \vee 1$, $f_I|_M = T$; neither a nor b

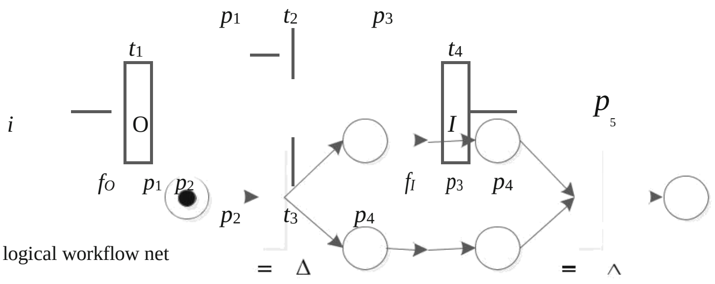

Fig. 1 A simple logical workflow net

in $a\ b = a \wedge b$ cludes a token, $f_I\,|M = F$. Suppose $f_O = a\ b$ and $f_O\,|M = T$, if both a and b need a token, $a\ b = 1 \wedge 1$; if a needs a token, b needs no token, $a\ b = 1 \vee 0$;if a needs no token, b needs a token, $a\ b = 0 \vee b$; If neither a nor b needs a token, $a\ b = 0 \vee 0$, $f_O\,|M = F$.

A simple logical workflow net is shown in Fig. 1

Definition 2 Firing rules of transitions in logical workflow nets.

(1) $t \in T_G$, $\forall p \in {}^{\bullet}t$, $M(p) \geq 1 \Rightarrow M[t>$. And after firing t, $\forall p \in t^{\bullet}$, $M'(p) = M(p) + 1$;

(2) $t \in T_O$, $\forall p \in {}^{\bullet}t$, $M(p) \geq 1 \Rightarrow M[t>$. And after firing t, only place p satisfying $f_O\,|M = T$, $M'(p) = M(p) + 1$, $\forall p \in {}^{\bullet}t$, $M'(p) = M(p) - 1$;

(3) $t \in T_I$, as long as $f_I\,|M = T$, $f_I\,|M = T \Rightarrow M[t>$. And after firing t, $\forall p \in t^{\bullet}$,

$M'(p) = M(p) + 1$, for place p satisfying $f_I\,|M = T$, $M'(p) = M(p) - 1$.

2.2 Soundness of Logical Workflow Nets

Definition 3 Let LWF_net $= (P, T, F, I, O, M_0)$, $M_0 = i$, $M_D = o$. LWF_net is sound if and only if

(1) $\forall M \in R[M_0, \sigma]$, $\exists \sigma$, $M[\sigma > \Rightarrow M_D$.

(2) $\forall t \in T$, $\exists M \in R[M_0, \sigma]$, $M[t>$.

For ordinary workfolw nets, if their net structures are same, their soundness are same. Different from ordinary workflow nets, even though net structures of logical workflow nets are same, their soundness is not certainly same. As is shown in Fig. 2, the two logical workflow nets have same net structures, but the left workflow in Fig. 2a is not sound and the right workflow in Fig. 2b is sound. The transition t_3 in Fig. 2a is subject to a logical expression $f_I = p_1 \wedge p_2$. If $M[t_3>$, then each of two places p_1 and p_2 in M in Fig. 2a must have a token. But it is impossible that p_1 and p_2 in Fig. 2a get a token at the same time, so t_3 is a dead transition and the left workflow in Fig. 2a is not sound. However, the transition t_3 in Fig. 2b is subject to a logical expression $f_I = p_1\ p_2$. As long as either of two places p_1 and p_2 in Fig. 2b have a token, $f_I = p_1\ p_2 = T$ and t_3 can be fired. Thus t_3 is live and the right workflow in Fig. 2b is sound.

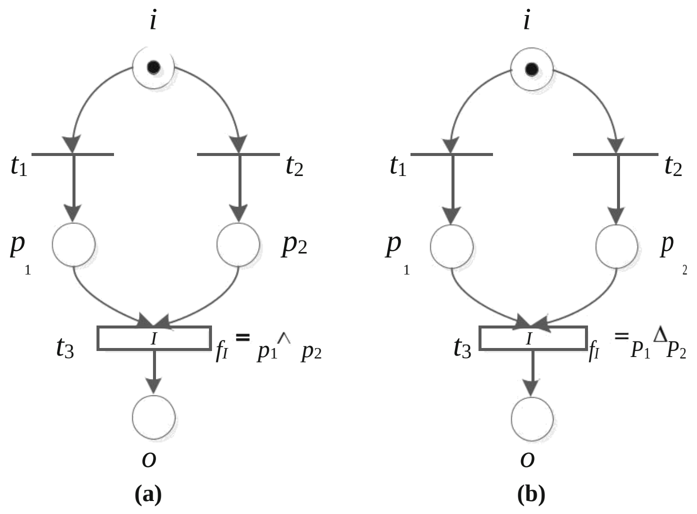

Fig. 2 Two logical workflow nets with same structure

3 Composed Logical Workflow Nets

The related concepts of composed logical workflow nets are introduced in this section.

3.1 The Definition of Composed Logical Workflow Nets

In reality, each organization is not isolated, Multiple organizations need to communicate each other. Each organization can be modeled by a logical workflow net. Communications of multiple organizations can be modeled by interface places. A logical workflow net model of multiple organizations can be built by linking each log-ical workflow net denoting each involved organizations with interface places denoting communications of different organizations. The resulting model is called a composed logical workflow net.

As is shown in Fig. 3, there are two logical workflow nets representing two organizations. One is shown in Fig. 3a, and the other is shown in Fig. 3b. Two organizations need to cooperate and coordinate to do business. The composed logical workflow net representing a cooperative business process of two organizations is described in Fig. 3c.

Places $p1$ and $p2$ are two interface places denoting communications of two organizations. The composed logical workflow net model of two organizations is constructed by linking two logical workflow nets in Fig. 3a, b with interface places based on a practically cooperative business process between two organizations.

Definition 4 A composed logical workflow net is a six-tuple CLWF_net = $(\overset{m}{\underset{i=1}{}} N_i,$ $P_C, I_C, O_C, F_C; M_{C0})$, where

(1) $N_i = \{P_i, T_i, I_i, O_i, F_i; M_i\}$ denotes the i th logical workflow net in CLWF_net;

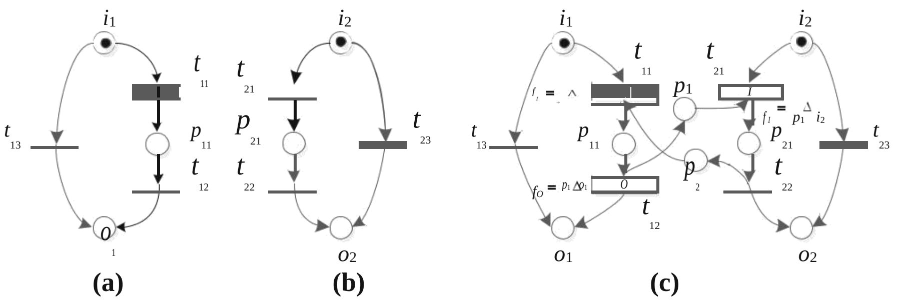

Fig. 3 Two logical workflow nets and the composed logical workflow net

(2) P_C denotes a interface place set, $\forall\ p \in P_C$, $|p^{\bullet} \cap t_i| = |{}^{\bullet}p \cap t_j| = 1$ $(t_i \in N_i, t_j \in N_j, i = j)$;

(3) I_C denotes a logical input expression set of transitions concerning interface places;

(4) O_C denotes a logical output expression set of transitions concerning interface places;

(5) $F_C \subseteq (P_C \times P_C^{\bullet}) \cup ({}^{\bullet}P_C \times P_C)$ denotes a directed arc set representing relations between interface places and transitions;

(6) $M_{C0} = i_1 + i_2 + \ldots + i_m$ denotes the initial marking in CLWF_net.

3.2 Soundness of Composed Logical Workflow Nets

Definition 5 Let CLWF_net = $(\bigcup_{i=1}^{m} N_i, P_C, I_C, O_C, F_C; M_{C0})$ be a composed logical workflow net, CLWF_net is sound if and only if

(1) $\forall M \in R[M_{C0}, \sigma], \exists \sigma, M[\sigma >\Rightarrow M_D$.

(2) $\forall t \in T, \exists M \in R[M_{C0}, \sigma], M[t>$.

In Fig. 4a, the composed logical workflow net is composed of two sound logical workflow nets in Fig. 4b, c. However, the composed logical workflow net in Fig. 4a is not sound, because transitions t_{11} and t_{21} are dead. $\neg\exists M \in R[M_{C0}, \sigma], M[t_{11}>$ or $M[t_{21}>$. A sound composed logical workflow net can be gotten on the premise of not changing net structure. If the logical input expression of the transition t_{21} is changed into $f_I = i_2\ p_1$, the composed logical workflow net becomes sound. Thus, soundness of composed logical workflow net is not only related to net structure but also logical expressions of logical transitions.

4 Soundness Determination of Logical Workflow Nets

Path nets and single line nets of logical workflow nets are proposed and the method of determining soundness of logical workflow nets is put forward in this section.

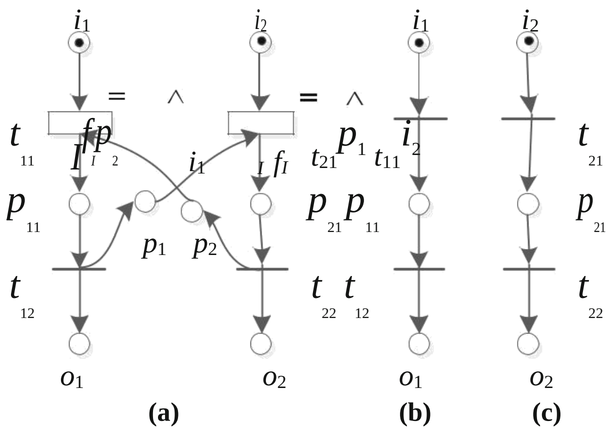

Fig. 4 An unsound composed logical workflow net

4.1 Path Nets and Single Line Nets

Definition 6 Let LWF_net $= (P, T, I, O, F; M_0)$, an acyclic net $PN = (P, T, I, O, F)$ is a path net of LWF_net, where

(1) $\forall t_i, t_j \in T (i = j), p \in P, \neg\exists \{t_i, t_j\} \in {}^{\bullet}p$ and $\neg\exists \{t_i, t_j\} \in p^{\bullet}$;

(2) $i, o \in P, t_i, t_j \in T, |i^{\bullet} \cap t_i| = |{}^{\bullet}o \cap t_j| = 1 (i = j)$;

(3) $t \in T, \int t \rightarrow t : t$ is a mapping of t in LWF_net, the preset and postset of t are as follows:

   (a) $t \in T_G$, the preset and postset of t are same as those of t;

   (b) $t \in T_I$, the postset of t is same as that of t.the preset of t will vary depending on the logical input expression.

      (i) $p \in {}^{\bullet}t$, if the logical input expression $f_I =< \exp >$ p, when $|{}^{\bullet}p| = 1$, $p \in {}^{\bullet}t$, the mapping of logical input function of t $f_I =< \exp > \wedge p$;when $|{}^{\bullet}p| = 0, p \notin {}^{\bullet}t$, the mapping of logical input function is $f_I =< \exp > \vee 0$;

      (ii) $p \in {}^{\bullet}t$, if $f_I =< \exp > \wedge\ p, p \in {}^{\bullet}t$.

   (c) $t \in T_O$, the preset of t is same as that of t.The postset place of t will vary depending on the logical output function.

      (i) $p \in t^{\bullet}$, if $f_O =< \exp > p$, when $|p^{\bullet}| = 1$, $p \in t^{\bullet}$,the mapping of logical output function is $f_O =< \exp > \wedge\ p$; when $|p^{\bullet}| = 0, p \notin t^{\bullet}$, the mapping of logical output function is $f_O =< \exp > \vee 0$.

      (ii) $p \in t^{\bullet}$, if $f_O =< \exp > \wedge\ p, p \in t^{\bullet}$.

(4) $F = \{(P \times T) \cup (T \times P)\} \cap F$.

Two path nets of the logical workflow net in Fig. 2b are shown in Fig. 5.

Definition 7 Let LWF_net $= (P, T, I, O, F; M)$, a acyclic net $SN = (P, T, I, O, F)$ is a single line net of LWF_net where

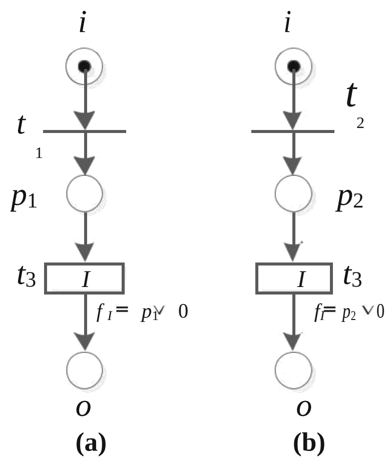

Fig. 5 Two path nets of the logical workflow net in Fig. 2b

(1) $\forall t_i, t_j \in T (i = j), p \in P, \neg\exists\{t_i, t_j\} \in {}^\bullet p$ or $\neg\exists\{t_i, t_j\} \in p^\bullet (i = j)$;
(2) $t \in T, |i^\bullet \cap t| = 1$;
(3) $t \in T$, $\int t \rightarrow t$, the preset and postset of t are as follows:

(a) $t \in T_G \cup T_O$, the preset and postset are same as those of t;
(b) $t \in T_I$, the postset of t is same as that of t.the preset of t will vary depending on the logical input function.

(i) $p \in {}^\bullet t$, if logical input function $f_I = \langle exp \rangle \wedge p$, when $|{}^\bullet p| = 1$, $p \in {}^\bullet t$, the mapping of the logical input function of t is $f_I = \langle exp \rangle$; when $|{}^\bullet p| \neq 0$, $p \notin {}^\bullet t$, the mapping of the logical input function of t is $f_I = \langle exp \rangle \vee 0$;

(ii) $p \in {}^\bullet t$, if $f_I = \langle exp \rangle \wedge p$, $p \in {}^\bullet t$.

(4) $F = \{(P \times T) \cup (T \times P)\} \cap F$.

All single line nets of the logical workflow net in Fig. 2b are shown in Fig. 6

4.2 Soundness Determination of Composed Logical Workflow Nets

Definition 8 A single line net *SN* is included by a path net *PN*, if all the transitions in a single line net *SN* are included in a path net *PN*.

Definition 9 A logical workflow net *LWF_Net* is covered by path nets if and only if each transition in the logical workflow net *LWF_Net* is included in at least one of all path nets.

Theorem 1 *A logical workflow net LWF_Net is sound if and only if each of its single line nets can be included by at least one of path nets, and the logical workflow net LWF_Net is covered by Path nets.*

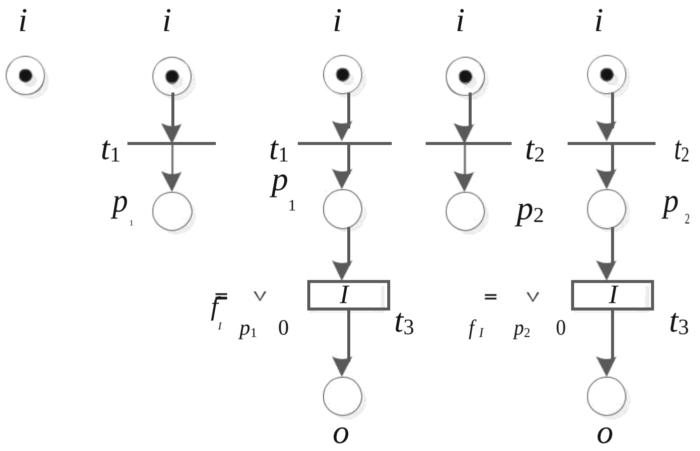

Fig. 6 Single line nets of the logical workflow net in Fig. 2b

Proof [Necessity] If a logical workflow net *LWF_Net* is sound, *LWF_Net* exists a single line net *SLN* and σ is a transition sequence of the single line net. Because *LWF_Net* is sound, there exist a complete σ , σ is a prefix of σ . Because σ must be contained by a path net *PN* of *LWF_Net*, so the single line net *SLN* is included by a path net *PN*. *LWF_Net* is sound, therefore, for each transition in *LWF_Net*, there exists a complete firing sequence σ , i [σ > o. Therefore *LWF_Net* is covered by its path nets.

[Sufficiency] If each of all Single line nets of logical workflow net *LWF_Net* is included by path nets and *LWF_Net* is covered by path nets. According to the definition of coverage, all transitions in *LWF_Net* are possible to be fired. That is, $\forall t \in T$, $\exists M \in R[M_0, \sigma]$, $M[t>$. If there exists a state *M*, in *M*, there exists a place *p* containing a token except the end place, and there exist no transition to be fired. *p* exists in a single line net, based on the premise, a single line net is covered by path nets. Therefore, there certainly exist transitions which can be fired, which is contrary
to the hypothesis.

There exists no path net in Fig. 2a, The net in Fig. 2b is covered by two path nets in Fig. 5a, b. Each single line net in Fig. 6 is included by one of path nets in Fig. 5. According to Theorem 1, it can be concluded that the logical workflow net in Fig .2b is sound.

4.3 Algorithm for Determining a Path Net of a Logical Workflow Net

The algorithm is given to derive path nets of logical workflow nets.

```
int main(){
N is a logical workflow net;
  N' =i ;
  If (Is_PathNet(N,N'))
  return 0;
}
 Is_PathNet(N,N'){
  Mc={p∈N', p•=Φ in N'};
  Tc={t∈N|Mc[t>};
  if(Mc≠O&Tc≠Φ){
     while(t∈Tc){
  if( t ∈ T_G ) N''= N'∪t ∪t• ∪(•t×t)∪(t×t•) ;
  else   N''= N'∪t ∪select(t,1)∪(select(t,0)×t)
∪(t×select(t,1));
    Is_PathNet( N , N'');
    }
}
    else if (Mc≠O&&Tc=Φ){
    return null;
}
return N';}
List select(t,flag){
List P;
f is the logical expression of t
if(t∈ T_I&flag==0){
          List Pc=•t;
          while(p∈Pc){
          If(f_I=exp∆p&|•p|=0 in N′) p∉  Pc ; }
          P=Pc;}
  else if(t∈T_0&flag==1){
      List   Pc=t•;
      While(p∈Pc){
      If (f0==<exp>∆p& |•p|=0 in N′) p  ∉  Pc ;}
       P=Pc;}
else {
    if(flag==1) P=t•;
    else P=•t;}
return P;}
```

5 Soundness Determination of Composed Logical Workflow Nets

Composed path nets and composed single line nets of composed logical workflow nets are proposed and the method of determining soundness of composed logical workflow nets is put forward in this section.

Definition 8 Let CLWF_net $= (\bigcup_{i=1}^{m} N_i, P_C, I_C, O_C, F_C; M_{C0})$, a acyclic net $N = (\bigcup_{i=1}^{m} N_i, P_C, T_C, I_C, O_C, F_C)$ is a composed path net of CLWF_net where

(1) $N_i = (P_i, T_i, I_i, O_i, F_i)$ is a path net of $N_i = (P_i, T_i, I_i, O_i, F_i; M_i)$;

(2) $T_i = \{T_{iG} \cup T_{iI} \cup T_{iO} \cup T_{iC}\}$, T_{iG} is a finite set of ordinary transitions, T_{iI} is a finite set of logical input transitions, T_{iO} is a finite set of logical output transitions, T_{iC} is a finite set of transitions connected to interface places;

(3) $P_C = \bigcup_{i=1}^{m} ({}^{\bullet}T_{icG} \cup T_{icG}{}^{\bullet}) \cup (C({}^{\bullet}T_{icI}) \cup T_{icI}{}^{\bullet}) \cup ({}^{\bullet}T_{icO} \cup C(T_{icO}))$ $(T_{icG}, T_{icI}$ and T_{icO} respectively denote the set of ordinary transitions, logical input transi-tions and logical output transitions connected to the interface places);

(4) $I_C = \int I_C, \int: f_I \in I_C$, if $p \in P_C$, when $f_I =< \exp >$ p, and $|{}^{\bullet}p| = 1$, then $\int$ $f_I =< \exp >$ p, otherwise, $\int f_I =< \exp > \vee 0$;

(5) $O_C = \int O_C, \int: f_O \in O_C, p \in P_C$, if $f_O =< \exp >$ p, and $|p^{\bullet}| = 1$, then $\int$ $f_O =< \exp > \wedge p$, otherwise $\int f_O =< \exp > \vee 0$;

(6) $F_C = \bigcup^{m} \{(P_C \times T_{ic}) \cup (T_{ic} \times P_C)\}$;

(7) $C(t)$ denotes the selection of contained place for different type of t;

(a) $t \in T_{icI}, \int t \to t$, the postset places of t is the same as those of t. The preset places of t $p = \{p | p \in {}^{\bullet}t, p \in P_C\}$ will vary depending on the logical input expression.

(i) $p \in P$, if $f_I =< \exp >$ p, when $|{}^{\bullet}p| = 1, p \in {}^{\bullet}t$; when $|{}^{\bullet}p| = 0, p \in/ {}^{\bullet}t$;

(ii) $p \in P$, if $f_I =< \exp > \wedge p, p \in {}^{\bullet}t$.

(b) $t \in T_{icO}$, the preset places of t is the same as those of t. The postset places of t $P = \{p | p \in t^{\bullet}, p \in P_C\}$ will vary depending on the logical output expression.

(i) $p \in P$, if $f_O =< \exp >$ p, when $|p^{\bullet}| = 1, p \in t^{\bullet}$; when $|p^{\bullet}| = 0, p \in/ t^{\bullet}$;

(ii) $p \in P$, if $f_O =< \exp > \wedge p, p \in t^{\bullet}$.

Definition 9 Let CLWF_net $= (\bigcup_{i=1}^{m} N_i, P_C, I_C, O_C, F_C; M_{C0})$, $N = (\bigcup_{i=1}^{m} N_i, P_C, I_C, O_C, F_C)$ is a composed single line net of CLWF_net where

(1) $N_i = (P_i, T_i, I_i, O_i, F_i)$ is a path net of $N_i = (P_i, T_i, I_i, O_i, F_i; M_i)$;

(2) $T_i = \{T_{iG} \cup T_{iI} \cup T_{iO} \cup T_{iC}\}$;

(3) $P_C = \{({}^{\bullet}T_{icG} \cup T_{icG}{}^{\bullet}) \cup (C({}^{\bullet}T_{icI}) \cup T_{icI}{}^{\bullet}) \cup ({}^{\bullet}T_{icO} \cup C(T_{icO}{}^{\bullet}))$;

(4) $I_C = \int I_C, \int: f_I \in I_C$, if $p \in P_C$, when $f_I =< \exp >$ p and $|{}^{\bullet}p| = 1$, $\int$ $f_I =< \exp >$ p, otherwise, $\int f_I =< \exp > \vee 0$;

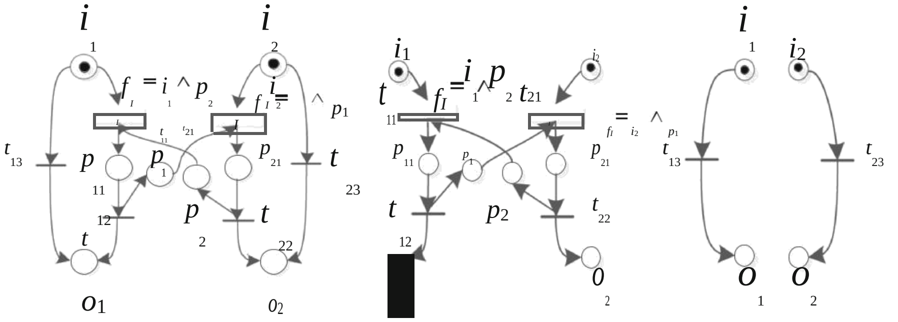

Fig. 7 **a** An unsound CLWF_net, **b** an incorrect composed path net, **c** a composed path net

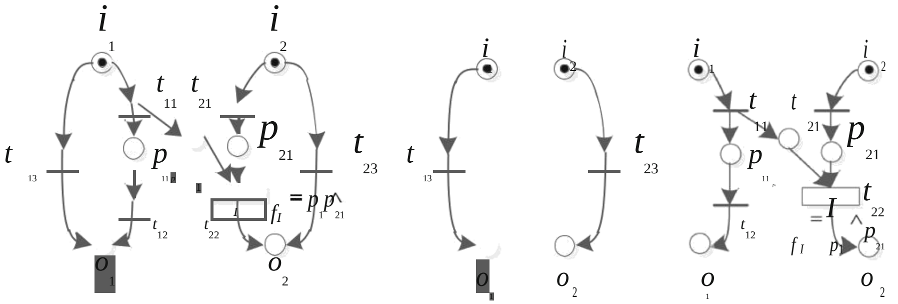

Fig. 8 **a** A sound CLWF_net, **b** a composed path net 1, **c** a composed path net 2

(5) $O_C = \int O_C, \int :\ f_O \in O_C$, if $p \in P_C$, when $f_O =< \exp >\quad p, \int f_O =< \exp >$ $\wedge\ p$;

(6) $F_C \subseteq \{(P_C \times T_{i\,C}) \cup (T_{i\,C} \times P_C)\} \cap F_C$.

Theorem 2 *A composed logical workflow net CLWF_net is sound if and only if each of its composed single line nets must be included by one of its composed path nets and CLWF_net is covered by its composed path nets.*

Proof [Necessity] A composed logical workflow net *CLWF_net* is sound, and σ is a transition sequence of a composed single line net. Because *CLWF_net* is sound, and thus there is a complete transition sequence σ , σ is the prefix of σ . Because *CLWF_net* is sound, a interface places as the output place of σ must be a output place of σ . Therefore a composed single line net is included in a composed path net. *CLWF_net* is sound, so all transitions in *CLWF_net* exist in at least one complete firing sequence. Therefore, *CLWF_net* is covered by composed path nets.

[Sufficiency]If *CLWF_net* is not sound, then there exist two cases: There exist dead transitions in *CLWF_net*; *CLWF_net* cannot reach the right end state. If there exist dead transitions in *CLWF_net*, and *CLWF_net* is covered by composed path nets, then each transition in *CLWF_net* exists at least one composed single line net. Therefore there exist no dead transition in *CLWF_net*. If *CLWF_net* cannot reach the right end state, that is, there exists a state *M*, there exist interface places containing a token in

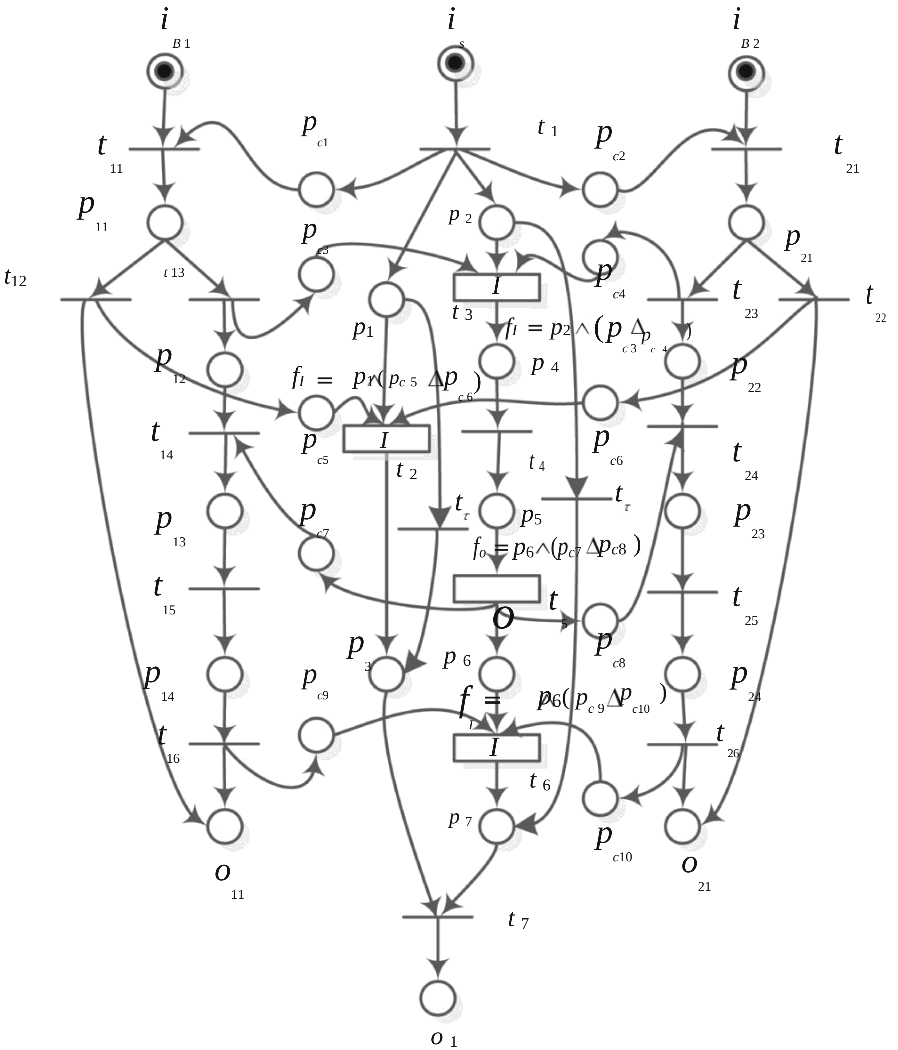

Fig. 9 A composed logical workflow net model of an e-commerce transaction process

M, but there exists no transition which can be fired. According to the premise that all the composed single nets in*CLWF_net* exist one of composed path nets, if interface places contain a token, there certainly exist a transition which can be fired, which is contrary to the hypothesis.

In Fig. 7, there exists only one composed path net shown in Fig. 7c in the composed logical workflow net shown in Fig. 7a. An incorrect composed Petri net is shown in Fig.7b. But the composed logical workflow net shown in Fig. 7a is not covered by the composed path net shown in Fig. 7c. So the composed logical workflow net shown in Fig. 7a is not sound. In Fig. 8, there exist two composed path nets shown in Fig. 8b, c in the composed logical workflow net shown in Fig 8a. And each of composed single

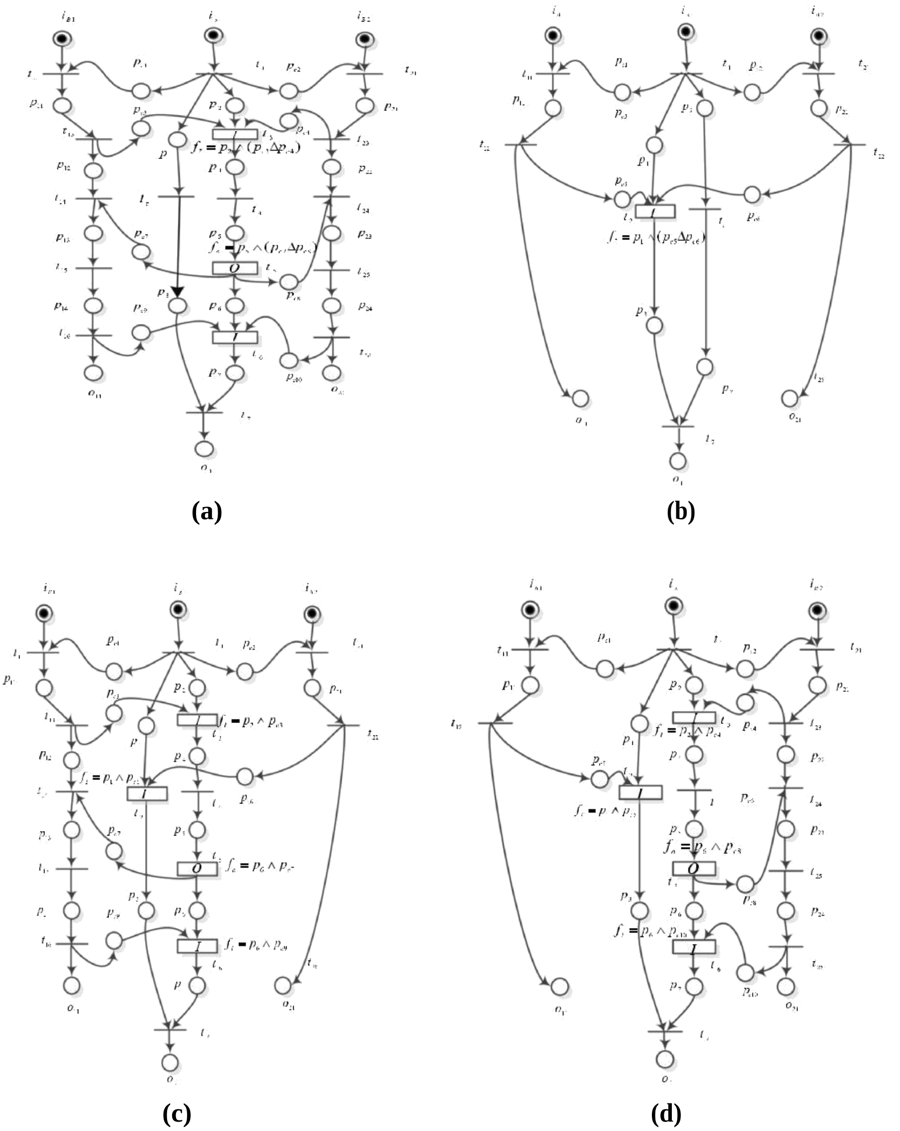

Fig. 10 Four composed path nets

line nets of the composed logical workflow net shown in Fig 8a is included by at least one of two composed path nets shown in Fig. 8b, c. So the composed logical workflow net shown in Fig. 8a is sound.

6 Soundness Analysis of an E-commerce Transaction Process Model

In e-commerce, a merchant needs to process orders from multiple users when they deal with transactions. The batch processing is a way to improve efficiency. A composed logical workflow net can better reflect the characteristics of batch processing in the workflow. Therefore, the composed logical workflow net is used to model the transaction process and judge soundness. A composed logical workflow net model of an e-commerce transaction process is shown in Fig. 9.

All composed path nets of are shown in Fig. 10a–d. It can be verified that each of all composed single nets is included in at least one of composed path nets in Fig. 10. And the composed logical workflow net of the e-commerce transaction process in Fig. 9 is covered by composed path nets. Thus, based on Theorem 2, the composed logical workflow net of the e-commerce transaction process in Fig. 9 is sound.

7 Conclusion

In this paper, we propose composed path nets and composed single line nets. By analyzing whether a composed logical workflow net is covered by its composed path nets and whether composed single line nets are included by composed path nets, soundness is judged. Finally, the methods are applied to soundness analysis of an e-commerce system modeled by a composed logical workflow net.

References

1. Liu, W., Wang, L., Du, Y.Y., Li, M.Z.: Deadlock property analysis of concurrent programs based on Petri net structure. Int. J. Parallel Prog. **45**(4), 879–898 (2017)
2. Van Der Aalst, W.M.P., Weske, M.: Reflections on a Decade of Interorganizational Workflow Research. Seminal Contributions to Information Systems Engineering, pp. 307–313. Springer, Berlin (2013)
3. Li, H.X., Du, Y.Y.: A survey of research issues and key technology for business process management. J. Shandong Univ. Sci. Technol. **34**(1), 22–28 (2015)
4. Tian, Y.H., Du, Y.Y.: A grouping algorithm of optimal alignments. J. Shandong Univ. Sci. Technol. **34**(1), 29–34 (2015)
5. Han, Y.B., Wang, J., Zhang, P.: Business-oriented service modeling: a case study. Simul. Model. Pract. Theory **17**(8), 1413–1429 (2009)
6. Wang, J., Du, Y.Y., Yu, S.X.: Coloured logic Petri nets and analysis of their reachable trees. Enterp. Inf. Syst. **9**(8), 900–919 (2015)
7. Du, Y.Y., Ning, Y.H.: Property analysis of logic Petri nets by marking reachability graphs. Front. Comput. Sci. **8**(4), 684–692 (2014)
8. Du, Y.Y., Qi, L., Zhou, M.C.: A vector matching method for analyzing logic Petri nets. Enterp. Inf. Syst. **5**(4), 449–468 (2011)
9. Liu, W., Du, Y.Y., Yan, C.: Soundness preservation in composed logical time workflow nets. Enterp. Inf. Syst. **6**(1), 95–113 (2012)
10. Liu, W., Du, Y.Y., Zhou, M.C., et al.: Transformation of logical workflow nets. IEEE Trans. Syst. Man Cybern. Syst. **44**(10), 1401–1412 (2014)
11. Hu, Q., Du, Y.Y., Yu, S.X.: Service net algebra based on logic Petri nets. Inf. Sci. **268**(6), 271–289 (2014)

12. Yan, C., Sun, H.T., Liu, W.: Study of fuzzy association rules and cross-selling toward property insurance customers based on FARMA. J. Intell. Fuzzy Syst. **31**(6), 2789–2794 (2016)
13. Li, Y.Q., Yan, C., Liu, W., Li, M.Z.: A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. Appl. Soft Comput. (2017). doi:10.1016/j.asoc.2017.07.027
14. Ding, Z.J., Wang, J.L., Jiang, C.J.: An approach for synthesis petri nets for modeling and verifying composite web service. J. Inf. Sci. Eng. **24**(5), 1309–1328 (2008)
15. Du, Y.Y., Jiang, C.J., Zhou, M.C., et al.: Modeling and monitoring of e-commerce workflows. Inf. Sci. **179**(7), 995–1006 (2009)
16. Sun, H.C., Du, Y.Y.: Soundness analysis of inter-organizational workflows. Inf. Technol. J. **7**(8), 1194– 1199 (2008)
17. Liu, W., Wang, P., Du, Y.Y., Zhou, M.C., Yan, C.: Extended logical Petri nets-based modeling and analysis of business processes. IEEE Access (2017). doi:10.1109/ACCESS.2017.2743113
18. Liu, G.J., Jiang, C.J., Wu, Z.H., et al.: A live subclass of Petri nets and their application in modeling flexible manufacturing systems. Int. J. Adv. Manuf. Technol. **41**(1), 66–74 (2009)
19. Liu, G.J., Reisig, W., Jiang, C.J., et al.: A branching process based method to check soundness of workflow systems. IEEE Access **4**, 4104–4118 (2016)