http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (6): 821-829, 2009
ISSN 1812-5638

Soundness Analysis of T-Restricted Interorganizational Logical Workflow Nets

[1]Wei Liu, [1,2]Yuyue Du and [1]Haichun Sun
[1]College of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266510, China
[2]State Key Laboratory of Computer Science, Chinese Academy of Sciences, China

Abstract: Interorganizational Logical Workflow Nets (ILWN) can efficiently model cooperative systems based on Petri nets, workflow techniques and temporal logic. But soundness of arbitrary ILWNs is hard to decide. This study defines the concept of T-restricted Logical Workflow Nets (LWN) and proposes an important subclass of ILWNs composed of n T-restricted LWNs: T-restricted ILWNs. The sufficient and necessary conditions of T-restricted ILWNs preserving soundness are obtained and the rigorous analysis approach is presented based on their static net structures only. Moreover, two approaches of combining n T-restricted LWNs into one T-restricted ILWN are given. The concepts and techniques proposed in this study are illustrated with a useful example of an auto gas station system.

Key words: Petri nets, logical workflow nets, soundness, liveness

INTRODUCTION

Due to the enhancement of reliability and safety of communication networks, the number of entities and the heterogeneity of cooperative systems are unprecedented and require new approaches to model and verify their correctness and temporal properties. It is inspiring that today's workflow management systems support business processes of complex systems via electronic networks and are widely used by organizations to coordinate the execution of various applications representing their day-to-day tasks. A workflow is a representation of a given process that consists of well-defined set of activities, referred to as tasks. Each of the tasks in the process represented by a workflow serves a given function and has some input requirements and may also generate information as a part of its output. The tasks in a workflow are usually related and dependent on one another. These task dependencies are called intra-workflow dependencies. Task dependencies may also exist across workflows where multiple organizations are involved in shared business processes, such task dependencies are referred to as inter-workflow dependencies. In general, task dependencies are divided into three types: control dependencies, value dependencies and external dependencies.

Van der Aalst (2000) defined interorganizational workflow nets (IOWF) to model loosely coupled interorganizational workflows. An IOWF-net describes the local workflows and the coordination structure for their interaction. XRL (eXchangeable Routing Language) language (Verbeek *et al*., 2002), which based XML language, was used for the specification of interorganizational workflows. Van der Aalst (2003) described formally the Public-To-Private (P2P) approach to construct interorganizational workflows based on a notion of inheritance. The public parts are related to each other, making up an interorganizational workflow. Each private workflow corresponds to an actual workflow as it is executed in one of the domains. The P2P approach guarantees that each private workflow is a subclass of the corresponding public part under projection inheritance. A new approach on the modeling of interorganizational workflows is presented based on nested Petri nets (Prisecaru and Jucan, 2008). It deals with loosely coupled Inter-organizational workflows: there are n local workflow processes which can behave independently, but need to interact at certain points in order to accomplish a global business goal. Soundness is introduced and proved.

An object-oriented technology is used to analyze the soundness of inter-organizational workflows (Sun and Du, 2008). It focuses on interactive messages in an inter-organizational system. Each workflow entity is packaged based on the object-oriented technologies. Each private workflow is considered as one object. The study put more attention on the whole part and interactive actions of the workflows. Some details inside one private workflow are neglected. It investigates interactive messages need to satisfy the properties in a sound inter-organizational workflow.

Corresponding Author: Wei Liu, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China

An Interorganizational Logical Workflow Net (ILWN) (Du *et al*., 2007) is presented for modeling and analyzing cooperative systems based on Logical Petri Nets (LPN), workflow techniques and temporal logic. It can model passing value indeterminacy and describe batch processing function of cooperative systems which the existing formal techniques cannot model. Through attaching logical expressions to some actions of an ILWN model, the size of the model is reduced. Thus, ILWN can mitigate efficiently the state explosion problem to some extent.

The modeling power of LPN and the equivalency between LPN and Petri Nets (PN) with inhibitor arcs (IPN) are analyzed and verified (Du and Guo, 2009). The equivalency is proved formally and a constructing algorithm of equivalent IPN from LPN is proposed based on the disjunctive normal forms of logic input/output expressions.

Soundness is an important property for ILWN. As a matter of fact, soundness can be decided; however, it is EXPSPACE hard (Van der Aalst, 2000). Thereby, soundness is discussed for an important subclass in this paper: T-restricted ILWN.

For the T-restricted ILWN, this study discusses how to compose ILWN to preserve soundness. Our results can be more expediently used by the designers of cooperative workflows in comparison with message sequence charts (Alur and Yannakakis,1999) because the method in this study can reduce the analysis complexity of ILWN consumedly. At last methods and concepts are illustrated with a useful example of an auto gas station system.

PRELIMINARIES

Here, we briefly review the notations of Petri nets, workflow nets and logical Petri nets.

A Petri net (Murata, 1989) is a four-tuple PN = (P, T, F, M_0), where P is a set of places, T is a set of transitions, $P\cap T=\phi$, $F\subseteq(P\times T)\cup(T\times P)$ are the edges (flow relations) and M_0: $P\rightarrow N\cup\{0\}$ (N is a natural number set) is the initial marking of PN. $R(M_O)$ represents the set of all markings reachable from M_O. $x\in P\cup T$: $\bullet x=\{y|\ (y, x)\in F\}$ and $x\bullet=\{y|\ (x, y)\in F\}$. For $Q\subset P\cup T$, $\bullet Q=\cup_{q\in Q}\bullet q$ and $Q\bullet=\cup_{q\in}q\bullet$.

Places are graphically drawn as circles, transitions are drawn as bars and the flow relations are drawn as directed arcs. A transition t is enabled if each place $p\in\bullet t$ contains at least one token. When an enabled transition t fires, for $\forall p\in\bullet t$, one token is removed from p and for $\forall p\in t\bullet$, one token is added to p. This results in a new marking M', denoted by M[t>M'.

A Petri net is strongly connected if and only if for every pair of nodes x and y, there is a path leading from x to y. It is bounded if and only if $\forall p\in P$, there is a natural number k such that $\forall M\in R(M_0)$: $M(p)\leq k$.

Definition 1: PN = (P, T, F, M_0) (Van der Aalst, 1999) is a workflow net (WF-net) iff:

- PN has two special places: i and o. Place i is a source place: $\bullet i=\phi$; Place o is a sink place: $o\bullet=\phi$
- If a transition $t^{\#}$ is added to PN which connects place o with i (i.e. $\bullet t^{\#}=\{o\}$ and $t^{\#}\bullet=\{i\}$), the resulting PN is strongly connected

Note that i (o) is used to denote place i (o) and the marking with only one token in place i (o).

A workflow net is constructed by several basic building blocks: AND-split And-Join, OR-split and OR-join (Van der Aalst, 2000).

However, it is difficult that the passing value indeterminacy and batch processing function of cooperative systems are described in WF-nets.

Definition 2: A Logical petri net (Du *et al*., 2007) is a 7-tuple LPN = (P, T, F, Dt, I, O, M), where P is a set of places; $T = T_D\cup T_I\cup T_O$ is a set of transitions; T_D is a set of delay transitions; T_I is a set of logical input transitions; T_O is a set of logical output transitions and P, T_D, T_I and T_O are disjunct sets; $F\subseteq(P\times(T_D\cup T_I\cup T_O))\cup((T_D\cup T_I\cup T_O)\times P)$ is a set of arcs; Dt is a function such that $\forall t\in T_D$, $Dt(t)\in R$, which denotes the delay time of t; I is a mapping function such that $t\in T_I$, I(t) is a logical input expression f_I, while O is a mapping function such that $\forall t\in T_O$, O(t) is a logical output expression f_O; M is a marking function.

The LPN representations of logical input and output transitions are shown in Fig. 1a and b.

Firing rules of the transitions in LPNs are as follows:

- $\forall t\in T_D$, $dt(t)=\tau$, t is said to be enabled if $\forall p\in\bullet t$; M(p)=1; t is said to be firable if its enabled time is equal to τ and firing t results in a new marking M': $\forall p\in\bullet t$: M'(p)=M(p)-1; $\forall p\in t\bullet$: M'(p)=M(p)+1
- $\forall t\in T_I$, $I(t)=f_I$, t is said to be enabled if $f_I|_M = .T.$, i.e., all input places of t satisfy the logical input expression f_I at M; if t is enabled, it can fire and firing t generates a new marking M': $\forall p\in\bullet t$: M'(p)=0; $\forall p\in t\bullet$: M'(p)=M(p)+1
- $\forall t\in T_O$, $O(t)=f_O$, t is said to be enabled if $\forall p\in\bullet t$, M(p)=1; if t is enabled, it can fire and generates a new marking M': $\forall p\in\bullet t$: M'(p)=M(p)-1; for $t\bullet$, $f_O|_{M'} = .T.$

A logical input transition is enabled only if the available tokens of its all input places satisfy a given

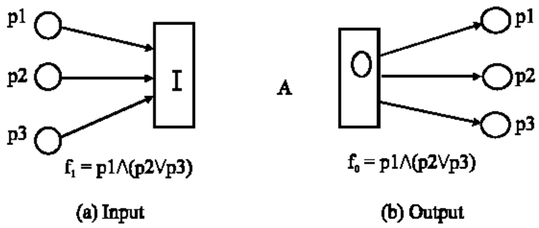

Fig. 1: Representations of logical (a) input and (b) output transitions

logical expression and there exists no token in its all input places after it fires. The notation f_I (f_O) is used to denote the logical expression of a logical input (output) transition. Logical input or output transitions are represented graphically by the rectangles in which mark I or O is embedded, respectively. Figure 1 shows that task A is enabled only if p1 has at least one token and p2 or p3 has at least one token. The enabling conditions of logical output transitions are the same as ones of the transitions in classical Petri nets. The output places of a logical output transition must satisfy its logical expression f_O and depends on new generated values when firing it. The output of logical output transitions is non-determinate in static structures. Figure 1b denotes that one token is generated in p1 and there is one token in one of p2 and p3, or in each of them, after executing task A.

Definition 3: Let M be a reachable marking in a LPN, α a firing sequence at M, f and g LPN formulae. $\alpha = \beta_i\gamma_j$, β_i is the prefix of α with length i and γ_j is the postfix of α excluding β_i, LPN formulas are defined recursively (Du *et al*., 2008).

- $<M, \alpha>|=p$ iff there is at least one token in place p at M
- $<M, \alpha>|=t_{fir}$ iff transition $t\in T$ is firable at M
- $<M, \alpha>|=t$ iff transition t fires, $t\in T_D\cup T_I\cup T_O$
- $<M, \alpha>|=f\vee g$ iff $<M, \alpha>|=f$ or $<M, \alpha>|=g$
- $<M, \alpha>|=f\wedge g$ iff $<M, \alpha>|=f$ and $<M, \alpha>|=g$
- $<M, \alpha>|=\neg f$ iff not $<M, \alpha>|=f$
- $<M, \alpha>|=f\Rightarrow g$ iff $<M, \alpha>|=f$ implies $<M, \alpha>|=g$
- $<M, \alpha>|=\bigcirc f$ iff $\alpha\neq\lambda$ and $<M_1,\gamma_1>|=f$
- $<M, \alpha>|=\Box f$ iff $<M_i,\gamma_i>|=f$ for every i: $0\leq i\leq |\alpha|$
- $<M, \alpha>|=\Diamond f$ iff $<M_i,\gamma_i>|=f$ for some i: $0\leq i\leq |\alpha|$
- $<M, \alpha>|=f\Delta g$ iff if $<M,\alpha>|=f$ and $<M,\alpha>|=g$ then $<M, \alpha>|=f\wedge g$, else $<M,\alpha>|=f\vee g$

Here, symbols $\neg$, $\wedge$ and $\vee$ are the Boolean connectives. Symbol Δ, as a substitutive operator, is used to describe the passing value indeterminacy. If f and g become true in M, then $f\Delta g$ is replaced by $f\wedge g$ and otherwise by $f\vee g$. In this study, .T. is used to represent a logical true value in logical expressions.

T-RESTRICTED ILWN

Definition 4: (T-1-live) A transition t in a Petri net (PN, M) is 1-live iff the following holds (Van der Aalst, 1999):

- For every marking M′ reachable without firing t, there is a state M′′ reachable from M′ which enables t
- For every marking M′ reachable via a firing sequence which fires t, there is no state reachable from M′ which enables t

Definition 5: A LWN=(P, T, F, DT, I, O, M) (Du *et al*., 2007) is called a logical workflow net iff:

- The LPN has two disjunct subsets, P_C and P_D and $P = P_C\cup P_D$, where P_C is a set of control places, P_D a set of interface data places
- P_C includes two special places: i and o. Place i is a source place: $\bullet i=\emptyset$. Place o is a sink place: $o\bullet=\emptyset$
- $\forall t\in T$, $\bullet t$ and $t\bullet$ include one control place at least, respectively
- If we add a transition $t^{\#}$ to the LPN which connects place o with i ($\bullet t^{\#}=\{o\}$ and $t^{\#}\bullet=\{i\}$), then its inner logical net $\overline{LN}=(\overline{P},\overline{T},\overline{F},DT,I,O)$ is strongly connected, where $\overline{P}=P_C$, $\overline{T}=T\cup\{t^{\#}\}$, $\overline{F}=F-(P_D\times T\cup T\times P_D)$

There are two types of places in LWN, control places and data places. Control places are used to deposit control tokens, data places to deposit data tokens. Data places are also called interface places.

Definition 6: An LWN (Sun and Du, 2008) is sound iff in $(\overline{LN},\overline{M}_0)$:

- $\forall\overline{M}\in R(\overline{M}_0)$, there exist a firing sequence α, such that $<\overline{M},\alpha>|=\Diamond(o\wedge(t^{\#}\Rightarrow o\overline{M}_0))$
- $\forall t\in\overline{T}$, $\exists\overline{M}\in R(\overline{M}_0)$, there exist a firing sequence α such that $<\overline{M},\alpha>|=\Diamond t$

In the above definition, requirement 1 means that for $\forall\overline{M}\in R(\overline{M}_0)$, there exists a firing sequence α such that $\overline{M}[\alpha>\overline{M}', \overline{M}'(O)=1$ and $\forall p\in\overline{P}-\{o\}:\overline{M}'(p)=0$. Requirement 2 means that there is no dead transition in the inner logical marking net $(\overline{LN},\overline{M}_0)$, $\forall t\in T$, $\exists\overline{M}\in R(\overline{M}_0)$, $\overline{M}[t>$. Since, the interface data places and their related arcs are omitted in an inner logical net, we can think of the marking net $(\overline{LN},\overline{M}_0)$ as its logical workflow process, i.e., the corresponding organization has full control over the local part of the LWN.

We demand that each interface data place in P_D has one sender and one receiver. Moreover, each transition involved in interface data places between instances has to be T-1-live. A logical workflow net which meets these two requirements is called a T-restricted logical workflow net.

Definition 7: (T-restricted logical workflow nets) A T-restricted logical workflow net is a logical workflow net which satisfies the following requirements:

- $p \in P_D$, $|\bullet p| = |p\bullet| = 1$
- $t \in \bullet P_D \cup P_D \bullet$, t is T-1-live

Definition 8: Let $LWN_j = (P_j, T_j, F_j, M_{0j}, DT_j, I_j, O_j)$ be an LWN of the j-th cooperative organization, j = 1, 2, . . .,n. ILWN = (P, T, F, M_0, DT, I, O) is an ILWN iff the following hold.

- $P = \cup_{1 \le j \le n} P_j$; $T = \cup_{1 \le j \le n} T_j$; $F = \cup_{1 \le j \le n} F_j$
- $t \in T_j$, $DT(t) = DT_j(t)$, j = 1, 2, . . .,n
- $t \in T_{Ij}$, $I_{(t)} = I_{j(t)}$ and $\forall t \in T_{Oj}$, $O(t) = O_{j(t)}$

Where, T_{Ij} is a set of logical input transitions in T_j and T_{oj} is a set of logical output transitions in T_j, j =1, 2, . . .,n.

- M_0 is the initial marking and $\forall p \in P_j : M_{0(p)} = M_{0j(p)}$, j = 1, 2, . . .,n
- The transition firing rules of the ILWN are the same as in Definition 3

If an additional transition $t^{\#}_j$ is added to LWN_j,which connects place o_j with i_j, j = 1, 2, . . ., n, $ILWN^{\#} = (P, T^{\#}, F^{\#}, M_0, DT, I, O)$ is called an extended ILWN, where $T^{\#} = T \cup \{t^{\#}_1, \ldots, t^{\#}_n\}$ and $F^{\#} = F \cup \{(o_1, t^{\#}_1), (t^{\#}_1, i_1), \ldots, (o_n, t^{\#}_n), (t^{\#}_n, i_n)\}$.

Definition 9: (T-restricted ILWN) A ILWN is called T-restricted ILWN if each of its LWNs is T-restricted.

An ILWN can be obtained through superposing the interface data places among the related LWNs by the definition if the local workflow processes of each cooperative organization are modeled by an LWN. However, the correctness of an ILWN may be destroyed due to asynchronous communication since the passing values between organizations may be subjected to order errors of some sending and receiving actions or incorrect choice of firing sequences. In Fig. 2, LWN1 and LWN2 are both sound, but their ILWN is not live, because t11, t12, t21 and t22 are dead transitions. But if the order of a receiving action and a sending action is exchanged in one of two LWNs, the ILWN is live. For example, if (p2, t21) and (t22, p1) are replaced with (p2, t22) and (t21, p1), respectively, the ILWN becomes live.

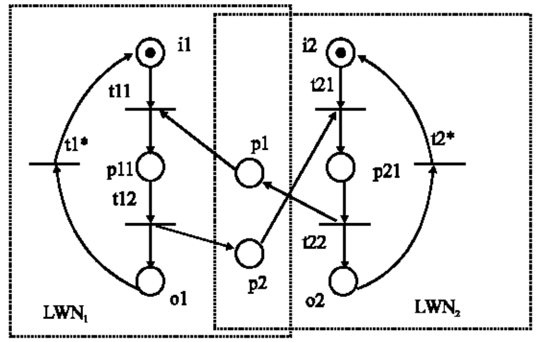

Fig. 2: An ILWN composed of two LWNs

Definition 10: An ILWN is sound (Du *et al*., 2007) iff

- Each of its LWNs is sound
- $ILWN^{\#}$ is live

Definition 11: Let LWN be a logical workflow net. C is a path leading from a node n_1 to a node n_k iff there is a node sequence $\langle n_1, n_2, \ldots, n \rangle_k$ in the LWN such that $(n_i, n_{i+1}) \in (T \times P) \cup (P \times T)$, i = 1,2,...,k-1. Assume that & (C) denotes the alphabet of a path C, i.e., & (C) = $\{n_1, n_2, \ldots, n_k\}$.

Soundness analysis of ILWNs: Here, soundness of T-restricted ILWN is analyzed. In the following, this study discusses soundness of a T-restricted ILWN composed of two T-restricted LWNs. The output transition of p is denoted by s(p). The input transition of p is denoted by r(p).

Theorem 1: Let ILWN be T-restricted and composed of two sound LWNs: LWN1 and LWN2. $\forall p, q \in P_D$, $s(p) \in T_1$, $s(q) \in T_2$, c1 is an arbitrary path leading from i_1 to s(p). c2 is an arbitrary path leading from i_2 to s(q).T_{i1} is the set of input transitions on path c1. T_{i2} is the set of input transitions on path c2.ILWN is sound iff $r(q) \notin T_{i1}$ or $r(p) \notin T_{i2}$.

Proof

Necessity: Since LWN_1 and LWN_2 are sound, each of their $\overline{LN}$ is live. If both LWNs in an ILWN are sound,the correctness of the ILWN depends on the liveness of the transitions related to the places in P_D and their precedence order in every local workflow process. Based on definition of ILWN, however, the order cannot be updated when an ILWN is constructed, i.e. the structural order of the transitions in every LWN is the same as in the ILWN. Consequently, when both LWNs are sound in an ILWN, if the ILWN is not live, this means that the incorrect order of the transitions related to the interface data places in some LWNs leads to some dead ones in the ILWN.

Case 1: If $t \in T \cap (\bullet P_D \cup P_D \bullet)$ is not in conditional routing structures, this means that the sending and receiving actions between LWN_1 and LWN_2 are only in sequential routing constructions and parallel routing constructions in LWN_1 and LWN_2. By means of the conditions of the theorem, $p \in P_D$, $|\bullet p| = |p \bullet| = 1$, $r(q) \notin T_{i1}$ or $r(p) \notin T_{i2}$, i.e., $r(q) \in T_{i1}$ and $r(p) \in T_{i2}$ are not satisfied simultaneously. That is to say, it will not happen that $s(p) \in T_1$ waits to receive the data q to be not sent by LWN_2 and $s(q) \in T_2$ waits to receive the data p to be not sent by LWN_1. Thereby, each transition in c_1 and c_2 is live in the ILWN.

Case 2: If $t \in T \cap (\bullet P_D \cup P_D \bullet)$ is in conditional routing structures, there exist two cases.

- For $t \in T \cap \bullet P_D$, if the branch that contains t is not selected, t will not fire. It is possible $r(T \bullet \cap P_D)$ will not fire. Thus, $r(T \bullet \cap P_D)$ is not live
- For $t \in T \cap P_D \bullet$, if the branch that contains t is not selected, t will not fire. It is possible $\bullet T \subseteq P_D$ still contains one token even when one workflow finishes

However, by means of the conditions of the theorem, $t \in \bullet P_D \cup P_D \bullet$, t is T-1-live, i.e.,$t \in \bullet P_D \cup P_D \bullet$ must fire once. So the above two cases will not happen. In addition, by means of the other conditions of the theorem, the analysis is similar to case 1.So $t \in T \cap (\bullet P_D \cup P_D \bullet)$ in conditional routing structures is also live in the ILWN.

Sufficiency: Let ILWN be sound, but $r(q) \notin T_{i1} \vee r(p) \notin T_{i2}$=.F., i.e., $r(q) \in T_{i1} \wedge r(p) \in T_{i2}$.This means that $s(p) \in T1$ waits to receive the data q to be not sent by LWN_2 and $s(q) \in T2$ waits to receive the data p to be not sent by LWN_1.This causes a deadlock. ILWN is not sound. This conclusion conflicts with conditions. So, the assumption is false, i.e., $r(q) \notin T_{i1}$ or $r(p) \notin T_{i2}$.

Corollary 1: An ILWN composed of n LWNs (LWN_1, LWN_2, ….. LWN_n) is sound if and only if:

- LWN_i, LWN_j($1<i,j<n$ and $i \neq j$). The ILWN' composed of LPN_i and LPN_j is sound
- ILWN composed of ILWN' and other n-2 LWNs (LWN_{m1}, LWN_{m2}, ….. $LWN_{m(n-2)}$ $m \notin i$ and $m \notin j$) is sound

Proof: It is similar to the Theorem 1. Based on Theorem 1,it is easy to verify the corollary 1.

Corollary 2: An ILWN composed of n LWNs (LWN_1, LWN_2, ….. LWN_n) is sound if and only if:

- LWN_i, LWN_j($1<i,j<n$ and $i \neq j$). The ILWN' composed of LWN_i and LWN_j is sound
- For every two LWNs :LWN_i, LWN_j, an $ILWN_m$ is obtained through the combination of them. Thus, n/2 ILWNs are gotten. The ILWN composed of these n/2 ILWNs is sound

Proof: It is similar to the Theorem 1.Based on Theorem 1, it is easy to verify the corollary 2.

THE EXAMPLE OF AN AUTO GAS STATION SYSTEM

An auto gas station example is modeled and analyzed to illustrate concepts and techniques proposed in this study. It is designed based on an auto gas station example (Corbett, 1996). The model is as follows:

1	Task body customer is	10	Task body Pump is	20	Task body operator is
2	Begin	11	Begin	21	Begin
3	Loop	12	Loop	22	Loop
4	Operator. Prepay;	13	Accept activate;	23	Select
5	Pump.Start;	14	Accept start;	24	Accept prepay do
6	Pump.Finish;	15	Accept finish do	25	Pump.Activate;
7	Accept change;	16	Operator.Charge;	26	End prepay;
8	End loop	17	End finish	27	Or
9	End customer	18	End loop	28	Accept charge do
				29	Customer.Charge;
				30	End charge;
				31	end select;
				32	End loop;

An auto gas station system is made up of three parts: a customer, an operator and a pump. In the example, an operator first accepts a customer's prepaid and then accepts charge of pump.

The workflow model of a customer is described by Fig. 3. The workflow model of an operator is shown in Fig. 4. The workflow model of a pump is shown by Fig. 5. In Fig. 3, some logical input expressions and logical output expressions must be satisfied: $f_o(t_3)$ = wait -ack-o-4∧ack-entry-o-4,$f_I(t_4)$=ack-accept-o-4∧ wait- ack-o-4, $f_o(t_5)$=wait-ack-p-5∧ack-entry-p-5, $f_I(t_6)$=ack -accept-p-5∧ wait-ack-p-5, $f_o(t_7)$=wait- ack-p-6∧ack- entry-p-6, $f_I(t_8)$=ack-accept-p-6∧ wait- ack-p-6∧ack -entry-c-29, $f_o(t_9)$=end-loop-8∧ack-accept-c-29. In Fig.4, some logical input expressions and logical output expressions must be satisfied: $f_I(t_{23})$=select-23 ∧ack-entry-o-4,$f_o(t_{24})$=ack-entry-p-25∧wait-ack-p-25, $f_I(t_{25})$=wait-ack-p-25∧ack-accept-p-25, f_o(t26)=ack- accept-o-4∧end-select-31, $f_I(t_{28})$=select-23∧ack-entry-o-16,$f_o(t_{29})$=ack-entry-c-29∧wait-ack-c-29,$f_I(t_{30})$= end-accept-30∧ack-accept-c-29. In Fig 4, some logical input expressions and logical output expressions must be satisfied: $f_I(t_{13})$=accept-13∧ack-entry-p-25,$f_o(t_{13}')$= ack-accept-p-25∧accept-14,$f_I(t_{14})$=ack-entry-p-5∧accept-

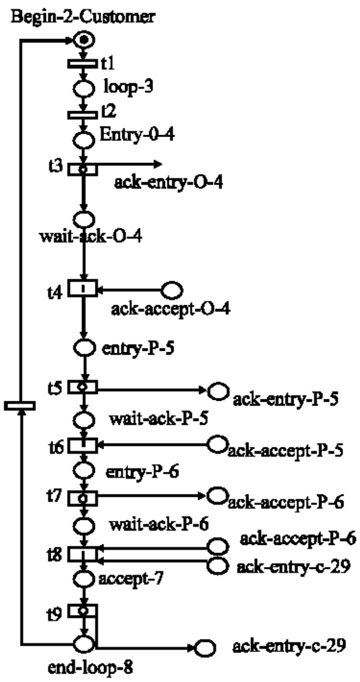

Fig. 3: The LWN model of a customer

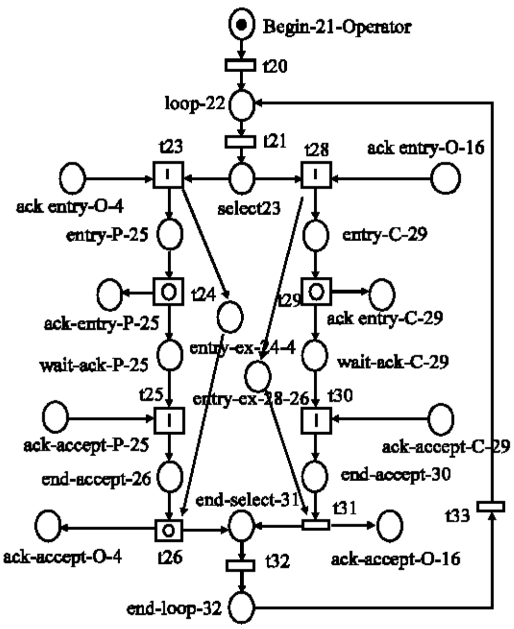

Fig. 4: The LWN model of an operator

14,$f_o(t_{14}')$=ack-accept-p-5∧ accep-15,$f_I(t_{15})$=accep-15∧ack-entry-p-6,$f_0(t_{16})$=ack-entry-o-16∧wait-ack-o-16,$f_I(t_{17})$= wait-ack-o-16∧ack-accept-o-16,$f_0(t_{18})$=ack-accept-p-6∧end-loop-18.

According to Definition 7, we can judge that three logical workflow nets in Fig. 3-5 are sound. In the following, how to combine the three LWNs into one

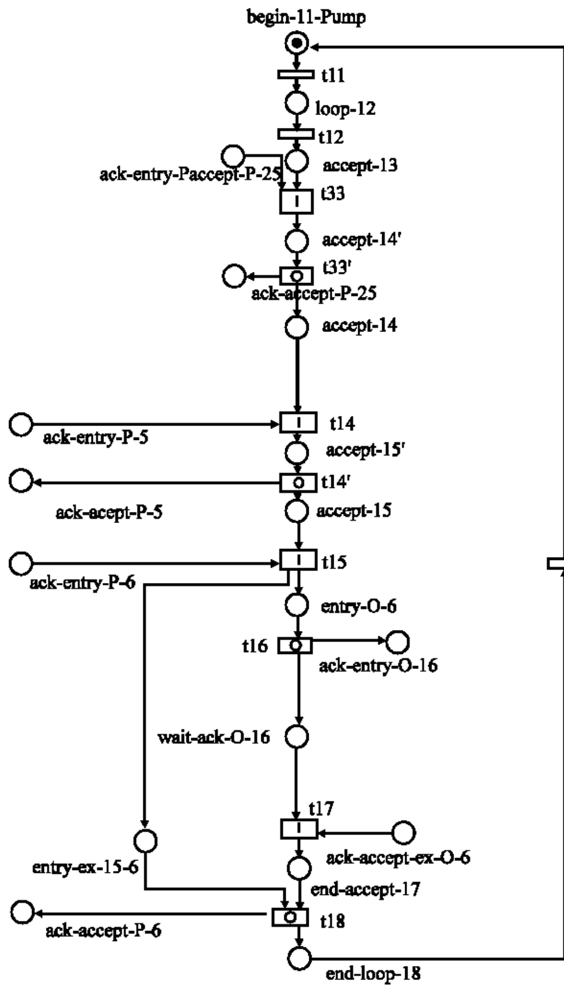

Fig. 5: The LWN model of a pump

ILWN and how to verify the soundness of the ILWN are illustrated.

Based on Corollary 1, firstly, the LWN of an operator and the LWN of a pump may be combined into one ILWN and the soundness of the ILWN can be verified by Theorem 1. Second, the LWN of an customer and the ILWN of an operator and an pump may be combined into one new ILWN and the soundness of the newly combined ILWN can be verified by Theorem 1:

(1) Combine the LWN of an operator and the LWN of a pump into one ILWN. There are four interface data places between the LWN of an operator and the LWN of a pump: let P_D={ack-entry-p-25,ack-accept-o-16,ack-accept-p-25,ack-entry-o-16}. s(ack-entry-p-25)=t24∈T(operator),s(ack-accept-o-16)=t31∈T(operator),s(ack-accept-p-25)=t13'∈T(pump),s(ack-entry-o-16)=t16∈T(pump).r(ack-entry-p-25)=t13∈T(pump),r(ack-accept-o-

16)=t17∈T(pump),r(ack-accept-p-25)=t25∈T(operator),r(ack- entry-o-16)=t28∈T(operator). there is one path c_{o1} from begin-21-operator to s(ack-entry-p-25)=t24.c_{o1}:begin-21-operator→t20→loop-22→t21→select-23→t23→ entry-p-25→t24.

There are four paths from begin-21-operator to s(ack-accept-o-16)= t31∈T(operator).c_{o2}:begin-21-operator→t20→loop-22→t21→select-23→t23→entry-p-25→t24→wait-ack-p-25→t25→end-accept-26→t26→end-select-31→t32→end- loop-32→loop-32→t21→select-23→t28→entry-c-29→t29→wait-ack-c-29→t30→ end-accept-30→t31.c_{o3}:begin-21-operator→t20→loop-22→t21→select-23→t23→ entry-ex-24-4→t26→end-select-31→t32→end-loop-32→loop-32→t21→select-23→t28→entry-c-29→t29→wait-ack-c-29→t30→end-accept-30→t31.c_{o4}:begin-21-operator→t20-loop-22→t21→select-23→t23→entry-p-25→t24→wait-ack-p-32→loop-32→t21→select-23→t28→entry-ex-28-16→t31.c_{o5}:begin-21-operator→t20→loop-22→t21→select-23→t23→entry-ex-24-4→t26→end-select-31→t32→end-loop-32→loop-32→t21→select-23→t28→entry-ex-28-16→t31.

There is one path from begin-11-pump to s(ack -ent ry -o-16)=t13'∈T(pump).c_{p1}: begin-11-pump →t11 → loop-12→t12→accept-13→t13→accept-14'→t13'.

There is one path from begin-11-pump to s(ack-entry-o-16)=t16∈T(pump).c_{p2}: begin-11-pump→t11→loop-12→t12→accept-13→t13→accept-14'→t13'→accept-14→t14→accept-15'→t14'→accept-15→t15→entry-o-16→t16. T_{o1} denotes the set of transitions receiving interface data places of & (c_{o1}). T_{o1}=∅.T_{o2} denotes the set of transitions receiving interface data places of & (c_{o2}). T_{o2}={t_{25},t_{28}}.T_{o3} denotes the set of transitions receiving interface data places of & (c_{o3}). T_{o3}={t_{28}}.T_{o4} denotes the set of transitions receiving interface data places of & (c_{o4}). T_{o4}={t_{25},t_{28}}.T_{o5} denotes the set of transitions receiving interface data places of & (c_{o5}). T_{o5}={t_{28}}.T_{p1} denotes the set of transitions receiving interface data places of & (c_{p1}). T_{p1}={t_{13}}.T_{p2} denotes the set of transitions receiving interface data places of & (c_{p2}). T_{p2}={t_{13}}.

r(ack-accept-p-25)=t25∈T(operator)∉T_{o1},r(ack-accept-p-25) = t25∈T(operator) ∈T_{o2}, r(ack-accept-p-25)=t25∈T(operator)∉T_{o3}, r(ack-accept-p-25)=t25∈T(operator) ∈T_{o4}, r(ack-accept-p-25) = t25∈T(operator) ∈T_{o5}.

r(ack-entry-o-16)=t28∈T(operator)∉T_{o1}, r(ack- entry - o-16)=t28∈T(operator) ∈T_{o2}, r(ack-entry-o-16) = t28 ∈T(operator) ∈T_{o3}, r(ack-entry-o-16) = t28∈ T(operator) ∈T_{o4}, r(ack-entry-o-16) = t28∈T(operator) ∈T_{o5}.

r(ack-entry-p-25) = t13∈T(pump)∈T_{p1},r(ack-entry- p-25) = t13∈T(pump)∈T_{p2}

r(ack-accept-o-16) = t17∈T(pump)∉T_{p1},r(ack-accept-o-16) = t17∈T(pump)∉T_{p2}

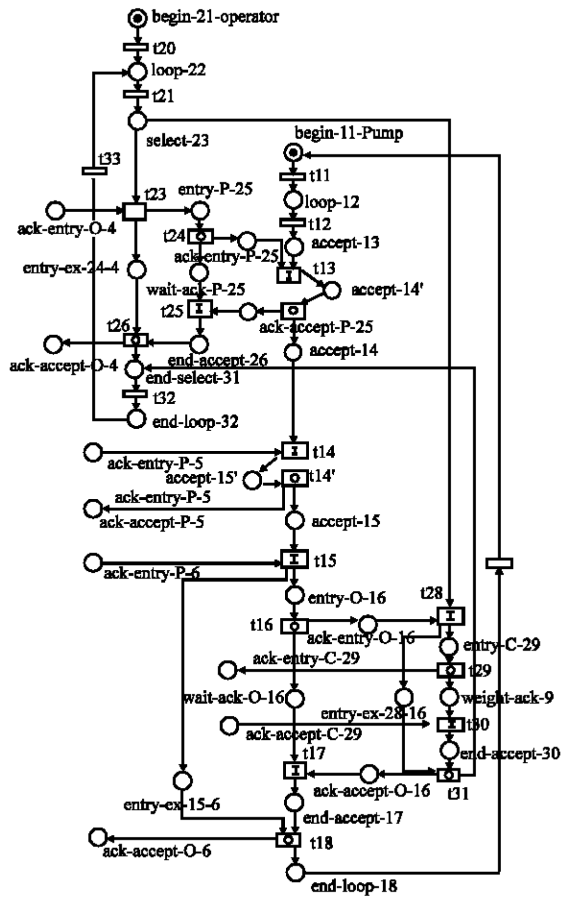

Fig. 6: The ILWN composed of an operator and a pump

For ack-entry-p-25∈P_D, ack-accept-p-26∈P_D, s (ack-entry-p-25) = t24∈T (operator), s(ack-accept-p-25) = t13' ∈T(pump),r(ack-entry-p-25)=t13∈T(pump)∈T_{p1},r(ack-accept-p-25)=t25∈T(operator)∉T_{01}.

In the same way, we can include for s (ack-entry-p-25) = t24∈T(operator), s(ack-entry-o-16) = t16∈T(pump). r(ack-entry-p-25) = t13∈T(pump)∈T_{p2}, r(ack-entry-o-16)= t28∈T(operator)∉T_{01}.

For s(ack-accept-o-16) = t31∈T(operator),s(ack-accept-p-25) = t13'∈T(pump). r(ack-accept-o-10) = t17∈ T(p ump)∈T_{p2},r(ack-accept-p-25) = t25∈T(operator)∉$T_{01,}$ r(ack-accept-o-10) = t17∈T(pump)∉T_{p1},r(ack-entry-p-25) = t25∈T(operator)∈T_{0i} = .T. i∈{2,3,4,5}.

For s(ack-accept-o-16) = t31∈T(operator),s(ack-entry-o-16) = t16∈T(pump). r(ack-accept-o-10) = t17∈T(pum p)∈T_{p2},r(ack-entry-o-16) = t28∈T(operator)∉$T_{01,}$ r(ack-accept-o-10) = t17∈T(pump)∉T_{p1},r(ack-entry-o-16) = t28∈T(operator)∈T_{0i} = .T. i∈{2,3,4,5}.

Thereby, the conditions of Theorem 1 are satisfied. According to Theorem 1, the soundness of the ILWN composed of an operator and a pump can be verified. The ILWN composed of an operator and a pump is shown in Fig. 6.

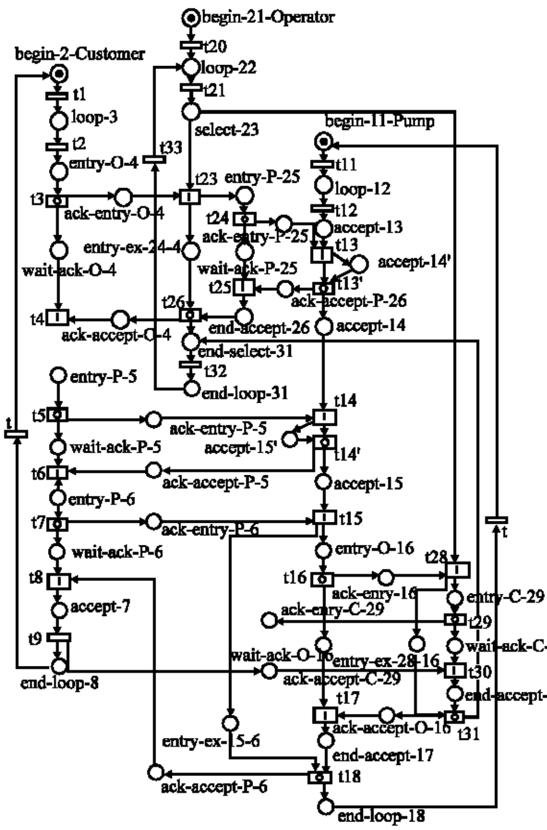

Fig. 7: The ILWN composed of a customer,an operator and a pump

(2) In the same way, combine the ILWN of a customer and the ILWN composed of an operator and a pump into one new ILWN. There are eight interface data places between the LWN of a customer and the ILWN composed of an operator and an pump: let P_D={ack-entry-o-4, ack-accept-o-4, ack-entry-p-5, ack-accept-p-5, ack-entry-p-6, ack-accept-p-6, ack-entry-c-29, ack-accept-c-29}.Based on Theorem 1, soundness of the ILWN can also be verified. The ILWN composed of a customer, an operator and a pump is shown in Fig. 7

So, according 1 and 2, it can be included that the ILWN composed of an operator, a pump and a customer is sound.

CONCLUSION

In this study, soundness of T-restricted ILWN is analyzed. A simple analysis approach is given and the inheritable conditions of soundness are obtained. Theorem 1 shows that an ILWN composed of two LWNs will preserve soundness if some conditions are satisfied. Corollary 1 and corollary 2 give the constructing algorithm of ILWN composed of n LWNs to ensure soundness. The use of the methods and techniques is demonstrated by analyzing the example of an auto gas station system.

Further research work is to study synthesis and property-preservation of other subclasses of ILWN and LPN.

For clarity of symbols, some symbols and abbreviations are listed in Table 1.

Table 1: Symbols and abbreviations

Symbol and abbreviation	Meaning
f_I	Logical expression of a logical input transition
f_O	Logical expression of a logical output transition
ϕ	Empty set
iff	If and only if
WN	Workflow nets
LPN	Logical Petri nets
IOWF	Interorganizational workflow nets
ILWN	Interorganizational logical workflow net
XRL	Exchangeable routing language

ACKNOWLEDGMENTS

This research is supported in part by the National Natural Science Foundation of China under Grants 60773034, 90818023, 90718012 and 60803032; the Taishan Scholar Construction Project of Shandong Province, China; the Scientific and Technological Developing Program of Shandong Province of China under Grant 2008GG30001024 and the Open Project of the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences under Grant SYSKF0804; the Research Project of SUST Spring Bud in Shandong University of Science and Technology in 2009 and the Graduate Innovation Fund of Shandong University of Science and Technology.

REFERENCES

Alur, R. and M. Yannakakis, 1999. Model checking of message sequence charts. Software Concepts Tools, 17: 70-77.

Corbett, J.C., 1996. Evaluating deadlock detection methods for concurrency software. IEEE Trans. Software Eng., 22: 161-180.

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2007. Modeling and analysis of real-time cooperative systems using petri nets. IEEE Trans. Syst., Man Cybernetics-Part A: Syst. Hum., 37: 643-654.

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2008. A petri nets based correctness analysis of internet stock trading systems. IEEE Trans. Syst. Man Cybern. C Applied Rev., 38: 93-99.

Du, Y.Y. and B.Q. Guo, 2009. Logic petri nets and equivalency. Inform. Technol. J., 8: 95-100.

Murata, T., 1989. Petri nets: Properties, analysis and applications. Proc. IEEE, 77: 541-580.

Prisecaru, O. and T. Jucan 2008. Interorganizational workflow nets: A petri net based approach for modelling and analyzing interorganizational workflows. Proceedings of the 4th International Workshop on Enterprise and Organizational Modeling and Simulation held in conjunction with the CAiSE'08 Conference, June 16-17, Montpellier, France, pp: 64-78.

Sun, H.C. and Y.Y. Du, 2008. Soundness analysis of inter-organizational workflows. Inform. Technol. J., 7: 1194-1199.

Van der Aalst W.M.P., 1999. Interorganizational workflows: An approach based on message sequence charts and Petri nets. Systems analysis-Modelling-Simulation, 35: 345-357.

Van der Aalst, W.M.P., 2000. Loosely coupled interorganizational workflows: Modeling and analyzing workflows crossing organizational boundaries. Inform. Manage., 37: 67-75.

Van der Aalst, W.M.P., 2003. Inheritance of interorganizational workflows: How to agree to disagree without loosing control?. Inform.Technol. Manage., 4: 345-389.

Verbeek, H.M.W., A. Hirnschall and W.M.P. van der Aalst, 2002. XRL/Flower: Supporting inter-organizational workflows using xml/petri-net technology. Proceedings of Web Services, E-Business and the Semantic Web, May 2002, Toronto, Canada, pp: 93-108.