A Workflow Decomposition Algorithm Based on Invariants*

PANG Shanchen¹², LIN Chuang¹, ZHOU Mengchu¹³ and LI Yin¹

¹Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
²College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 271019, China
³Department of Electrical Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA

Abstract — An efficient decomposition algorithm for a workflow model based on Petri net invariants is presented in this paper. The algorithm is a new kind of simplification strategy that decomposes a complex and sound Workflow net (WF-net) to a class of simple subnets that are able to describe the business cases. The existence of T-invariants and the coverage sets of WF-net transitions are analyzed and verified in detail. The advantages of the algorithm include its simplicity, and avoidance of state space explosion. It can overcome the shortage of the methods based on a depth traversal algorithm. Furthermore, it is readily comprehensible, and can be extended easily to allow parallel processing. The usability of the research results is illustrated by an example.

Key words — Workflow-net, Invariant, Petri nets decomposition algorithm.

I. Introduction

As a key for workflow management to achieve high efficiency, the methods of workflow modeling, analysis and optimization have attracted many researchers[1]. Petri nets are well-known formalism for describing concurrent discrete event dynamic systems[2,3,6,26,27]. Abundance of qualitative and quantitative analysis techniques, formal semantics, and local state-based system description, are three good reasons to use a PN-based workflow management system. There are various types of PN used for workflows, e.g., Time Petri net (TPN)[4–6], Colored Petri net (CPN)[7–9], Stochastic Petri net (SPN)[10–14], Fuzzy Petri net (FPN)[15–17], and Hybrid Petri net[19,20]. Most of researches focus on the process soundness and workflow performance.

According to Ref.[21], Workflow net (WF-net) can be used successfully to describe a workflow system process at logical levels. A simplification strategy that makes use of the equivalence mechanism of behavior and functionality has been applied to many fields successfully, e.g., Flexible manufacturing systems[26], Command and control systems[29], and Business process reengineering. It can be used to reduce a complex structure to a simple one to analyze workflow performance.

Li et al.[6] present a behavioral expression analysis method to provide a new method to evaluate workflow performance. It avoids efforts to find all states. Since it is a depth traversal process, it suffers from high time complexity. Aalst[21] discussed the relation between P-coverability and soundness, and proved a sound well-structured WF-net is P-coverable. Aalst conjecture there is a high correlation between P-coverability and soundness, but he did not show their relation completely.

This paper presents a necessary and sufficient condition for the soundness which shows the relation completely between P-coverability and soundness. And presents a decomposition algorithm based on Petri net invariants and aims to overcome the above shortcomings: Loss of resource information, and explosion of a state space. The proposed algorithm is a new simplification strategy that decomposes a complex and sound WF-net to simple subnets that well describe business cases.

First, this paper identifies and verifies T-invariants and the coverage sets of WF-net transitions. It then presents a new necessary and sufficient condition for the soundness of WF-net. Compared with other algorithms, the proposed one can overcome the shortage of the depth traverse and reduce decomposition complexity of a WF-net. It is simple, easily understandable and implementable. Its usability of the results is illustrated by an example.

The rest of this paper is organized as follows: Section II introduces Petri nets and workflow models. Section III studies the existence of T-invariants and the coverage sets of WF-net transitions. It also presents the algorithm based on T-invariant decomposition. Section IV gives an example. Section V summaries the related work. Finally, Section VI concludes the paper.

II. Workflow Model Based on Petri Nets

W.M.P. van der Aalst[21] shows that workflow primitives can be mapped onto Petri nets, and presents a workflow model based on Petri nets, which is called a Workflow net (WF-net). In order to use it for the rest of this paper, we brief its basic concepts. For a review of Petri nets, please refer to Refs.[2,
22, 26, 27]. We use \( L(M_0) \) and \( L(M) \) to denote all firable sequences of transitions starting at \( M_0 \) and \( M \).

**Definition 1** A Petri net \( PN = (P, T, F, M_0) \) is a workflow net if:

1. \( PN \) has two special places: \( i \) and \( o \). Place \( i \) is a source place, i.e., \( i^* = \emptyset \). Place \( o \) is a sink place, i.e., \( o^* = \emptyset \).
2. If we add a transition \( t^* \) to \( PN \), which connects place \( o \) with \( i \) (i.e., \( i^* = \{o\} \cap t^* = \{i\} \)), then the resulting extended net \( \tilde{PN} \) is strongly connected.
3. \( M_0(i) \neq 0 \) and \( M_0(o) = 0 \), \( \forall p \neq i \).

**Definition 2** A WF-net \( PN = (P, T, F, M_0) \) is sound if:

1. For every state \( M \) reachable from initial state \( M_0 \), there exists a firing sequence leading from state \( M \) to state \( M_f \) at which \( M_f(o) \neq M_0(i) \) and \( M_f(p) = 0 \), \( \forall p \neq o \). Formally:

\[
\forall M \in R(M_0), \exists \tau \in L(M) \Rightarrow M[\tau > M_f).
\]
2. State \( M_f \) is the only state reachable from state \( M_0 \) with at least one token in place \( o \). Formally:

\[
\forall M \in R(M_0) \wedge M \geq M_f) \Rightarrow (M = M_f).
\]
3. There are no dead transitions in WF-net. Formally:

\[
\forall t \in T, \forall M \in R(M_0), \exists M' \wedge M'' \in R(M), \exists M'[t > M''].
\]

**Definition 3** A Petri net \( PN \) is well-handled iff for any pair of nodes \( x \) and \( y \) such that one of the nodes is a place and the other a transition and for any pair of elementary paths \( C_1 \) and \( C_2 \) leading from \( x \) to \( y \), \( \alpha(C_1) \cap \alpha(C_2) = \{x, y\} \Rightarrow C_1 = C_2 \). A WF-net \( PN \) is well-structured if the extended net \( \tilde{PN} \) is well-handled.

In this paper, we assume that a WF-net \( PN \) is well-structured.

**Lemma 1** A WF-net \( PN \) is sound if \( \tilde{PN} \) is live and bounded.

**Lemma 2** A Petri net \( PN \) is conservative iff the places set \( P \) is a support of \( P \)-invariant.

**Lemma 3** A Petri net \( PN \) is structurally bounded if it is conservative.

**Lemma 4** A reversible Petri net \( PN \) is live iff every transition in it can fire once at least from the initial marking.

### III. Property Analysis and Decomposition of a WF-net

**Theorem 1** Let \( PN \) be a sound WF-net. For the extended net \( \tilde{PN} \), there exists at least one T-invariant.

**Proof** By Definition 2 and the definition of \( \tilde{PN} \), \( \exists \sigma = t_1t_2 \cdots t^n, M_0[\sigma > M_0] \), where \( \sigma \) is a friable transition sequence. Hence, we make an \( n + 1 \) dimension vector \( X = (x_1, x_2, \cdots, x_n, x_{n+1}) \), where \( x_i = |t_i|, x_{n+1} = |t_n^*|, i = 1, \cdots, n \), and \( |t_n^*| \) is the number of \( t_n \)'s appearances in friable transition sequence \( \tau \). By the state equation, \( A^T X = 0 \). By the definition of T-invariant, \( X \) is one T-invariant of \( \tilde{PN} \). Let \( T_i \) be the transition set of \( \tau \). By the state equation, \( X \) is the support of \( T_i \), and \( t \in T_i \). Since there is a contradiction with our assumed condition \( \forall \nu \in T, t \notin T_i \). Consequently, the transition set \( T \) is covered by the T-invariant support set.

**Theorem 2** Let \( PN \) be a sound WF-net. The output subnet generated by a minimal T-invariant support of \( \tilde{PN} \) is a T-Component net, and \( \tilde{PN} \) is covered by them.

**Proof** We assume that an output subnet \( PN' = (P', T', F') \) of \( \tilde{PN} \) as generated by a minimal T-invariant support is not a T-Component net. Then \( \exists p' \in P' \) such that \( |p'| \geq 2 \). Assume that \( |p'| = 2 \) with \( t_1, t_2 \in T' \). Since \( \tilde{PN} \) is well-structured, \( \exists p' \in P', p' \) pairs with \( p \), and \( |p'| \geq 2 \). If not, there is only one sink place \( o \), \( \exists p' \in P', p' \) pairs with \( p \), and \( |p'| = 1 \), and there are two paths form \( p \) to \( *p \) at least. Let one path be \( C_1 \) passing \( t_1 \), and the other path be \( C_2 \) passing \( t_2 \). By the definition of a well-structured \( \tilde{PN} \), \( \alpha(C_1) \cap \alpha(C_2) = \{p', *p\} \Rightarrow C_1 = C_2 \), i.e., \( t_1, t_2 \). But it is in contradiction with \( |p'| = 2 \). Consequently \( \exists p' \in P', |p'| \geq 2 \), where \( p' \) pairs with \( p \).

Let \( t_3, t_4 \in *p' \) be respectively in \( C_1 \) and \( C_2 \). Since \( PN \) is sound, \( \forall M \in R(M_0), \exists M' \in R(M) \), \( M'_t[t > M'_f > M_0, \forall M \in R(M_0), \exists M' \in R(M), M'_t[t > M'_f[t > M_0] \), then the path from \( t_3 \) to \( t_4 \) is still a T-invariant support. This is a contradiction with the definition of the minimal T-invariant support. Consequently, the output subnet of \( \tilde{PN} \) generated by a minimal T-invariant support is a T-Component net. By Theorem 2, \( \tilde{PN} \) is covered by its output subnets.

**Theorem 4** Let \( PN \) be a sound WF-net. There exists at least a P-invariant.

**Proof** We assume that there does not exist any P-invariant in \( PN \). Since the extended net \( \tilde{PN} \) is strongly connected, there exists a path from the source place \( i \) to the sink place \( o \). Let \( C = (i, p_1, \cdots, p_k, o) \) that is called a place path. \( \forall p_1 \in C \setminus \{i, o\} \), \( |p_1| \geq 2 \), by the proof of Theorem 3, \( \exists p_j \in C \setminus \{i, o\}, |p_j| > 2 \) where \( p_j \) pairs with \( p_1 \). Two cases are discussed as follows:

1. If \( p_1 \) is located in the path from \( p_1 \) to the sink place \( o \) of \( C \), then \( p_1 \) and \( p_j \) are respectively the or-split place and or-join place of a choice route in \( PN \). Hence, let all places in another path \( C_1 \) from \( p_1 \) to \( p_j \) merge into \( C \). We have a new set \( C = C \setminus C_1 \setminus \{t \in C \} \).

2. If \( p_j \) is located in the path of \( C \) from \( i \) to \( p_j \), then the path from \( p_1 \) to \( p_j \) is a cycle. As case (1), let all places in another path \( C_2 \) from \( p_1 \) to \( p_j \) merge into \( C \). We have a new set \( C = C \setminus C_2 \setminus \{t \in C \} \).

Repeat the above steps, until all the places that satisfy \( |p| \in P \geq 2 \) in \( C \) are traveled.

We need to prove that all the places in \( C \) assemble a P-invariant support set of \( PN \).

We suppose that a token flows from the source place \( i \) to the first place of a choice route or a cycle route, and is called \( p_1 \). Then there may exist two kinds of routes between \( i \) and
If it is a sequence route, when the token arrives in $p_i$, the number of the tokens is not changed through the path. If it is a concurrent route, $C$ is constructed in a way that one sequence path of the concurrent route is selected as the path from $i$ to $p_i$. The number of the token does not change when it traverses this path and the token eventually arrives at $p_i$. Whether $p_i$ is the first place of a choice or cycle route, the token in $p_i$ only flows into one of its successive places (marked as success ($p_j$)). Thus the number of tokens in $C$ does not change, and remains as a constant.

Through the above process, we can gain a nonzero m-vector $Y$ of integers. Clearly $AY = 0$, where $y_i$ is token increment or decrement in the $i$th place, $i = 1, 2, \cdots, m$. By the support set of P-invariant definition, all of the places in $C$ assemble a P-invariant support set of $PN$. There is a contradiction with our assumed condition. Consequently, there exists one P-invariant of $PN$ at least.

**Theorem 5** Let $PN$ be a sound WF-net. The place set $P$ is covered by the P-invariant support set $P = \{P_1, P_2, \cdots, P_k\}$ of $PN$.

**Proof** Assume that the place set $P$ is not covered by the P-invariant support set of $PN$. Then there exists a place $p \in P$, such that $\forall P_i \in P, p \notin P_i$. Since the extended net $\overline{PN}$ is strongly connected, there exists a path from the source place $i$ to the sink place $o$. Let $C = \{i, p_1, \cdots, p_k, o\}$ which is called a place path, $\forall p_i \in C \setminus \{i, o\}$ and $p \in C \setminus \{i, o\}$. Using the method of Theorem 4, we can construct a new set $C$, and all of the places in $C$ including $p$ assemble a P-invariant support of $PN$. This is a contradiction with our assumed condition. The conclusion is proved.

**Theorem 6** Let $PN$ be a sound WF-net. The outface subnet generated by the minimal P-invariant support is a P-Component net, and $PN$ is covered by them.

**Proof** By Theorem 4, it is easy to prove that the outface subnet generated by the P-invariant support is a P-Component net. By Theorem 5, it is easy to prove that $PN$ is covered by its outface subnets.

**Theorem 7** A well-structured WF-net $PN$ with $M_0$ being its home state is sound, iff (1) the outface subnet generated by the minimal T-invariant support of $\overline{PN}$ is a T-Component net, and $PN$ is covered by them, and (2) the outface subnet generated by the minimal P-invariant support is a P-Component net, and $PN$ is covered by them.

**Proof** We only prove the soundness equivalence definition of WF-net $PN$.

$(\Rightarrow)$ By Theorem 3, Theorem 6 and the definition of home state, it is easy to prove the conclusion.

$(\Leftarrow)$ First, we prove that $\overline{PN}$ is bounded. Because of the outface subnet generated by the minimal P-invariant support is a P-Component net, and $PN$ is covered by them, by Lemmas 2 and 3, $\overline{PN}$ is bounded.

Secondly, we prove that $\overline{PN}$ is live. Since the outface subnet generated by a minimal T-invariant support of $\overline{PN}$ is a T-Component net, and $\overline{PN}$ is covered by them, thus $\|X\| = T$ is one T-invariant support of $\overline{PN}$, i.e., $\overline{PN}$ is a repetitive net. It is known that $M_0$ is a home state, and $\overline{PN}$ is well-structured. We prove that $\overline{PN}$ is live [0] as follows.

We assume that $\overline{PN}$ is not live, i.e., $\exists t \in T$ is a dead transition. Then $\exists p \in ^* t, M \in R(M_0), M(p) = 0$, i.e., $^*p$ cannot fire. Repeat the above steps, we can discover a $p'$ finally. $^*p'$ can fire, and $|p'| \geq 2$. There are two cases. One is that there are at least two paths from $p'$ to $t$, it is a contradiction with the well-structured definition of $\overline{PN}$. The other is that $t$ is in one path of the choice routes from $p'$. Then if the token flows to $^*t, t$ can fire. It is a contradiction with our assumption. Therefore $\forall t \in T, t$ can fire once at least. By Lemma 4, $\overline{PN}$ is live.

Theorem 7 not only proves that a sound workflow net model can be decomposed to a group of T-Components or P-Components, but also presents the soundness verification method for a workflow net.

2. The decomposition algorithm based on T-invariants

Based on the prior results, we give an efficient workflow net decomposition algorithm based on T-invariants as follows.

**Algorithm 1. Decomposition algorithm based on T-invariants**

**Step 1** Construct the extended net $\overline{PN}$ by adding a new transition $t^*$ between the source place $i$ and sink place $o$, i.e., $^*t = \{o\}, t^* = \{i\}$.

**Step 2** Solve the equations $A^T X = 0$ with constraints $X > 0$ to obtain a group of minimal T-invariants $X_1, X_2, \cdots, X_1, T$ is their support set, and cover the transition set $T$. $T^*$ is the support set that contains transition $t^*$, denoted by $T^* = \{T_1, T_2, \cdots, T_k\}$. $T'' = T \setminus T^*$.

**Step 3** Construct their outface subnet of $\overline{PN}$ for $\forall T_j \in T^*$, denoted by $PN_j = \{p_i, T_j, F_j\}$. Let $T''_i = \{PN_1, PN_2, \cdots, PN_k\}$.

**Step 4** If $T'' = \emptyset$, then $T''_i = \{PN_1, PN_2, \cdots, PN_k\}$, and go to Step 5 directly. Otherwise, $\forall T_j \in T''$, if $\exists PN_i \in T''_i$, makes $T_i \cap T_j \neq \emptyset$ and $\exists p \in PN_i, t_j \in T_j$, such that $^*t_j = p$. Then let the outface subnet of $T_j$ in $\overline{PN}$ merge into $PN_i$, and obtain a new $PN_i$. Repeat this step until $T'' = \emptyset$.

**Step 5** Delete $t^*$ and its corresponding arcs $(o, t^*)$ and $(t^*, i)$ from each subnet $PN_j$.

IV. Instantiation and Complexity

Analysis

Li et al.[6] presents a WF-net model. Based on it, a new WF-net model is set up with an additional cycle (Fig.1). Fig.2 gives its extended version. We verify Algorithm 1 in terms of the model decomposition, and analyze its complexity and practicality as follows.
It is easy to prove that the WF-net PN (Fig.1) is sound. By Fig.2, we can have the incidence matrix $A$ of the extended net $PN$:

$$A = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

By solving equations $A^TX = 0$, we find the minimal T-invariants: $X_1, X_2$, and $X_3$, with their supports $T_1, T_2$, and $T_3$ covering the transition set $T$.

$$X_1 = (1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1);$$

$$T_1 = (t_1, t_2, t_4, t_6, t_7, t_8, t_{10}, t_{11}, t_{12}, t_{15}, t_{16});$$

$$X_2 = (1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1);$$

$$T_2 = (t_3, t_5, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{15}, t_{16});$$

$$X_3 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);$$

$$T_3 = (t_{10}, t_{13}, t_{14}).$$

By Step 2, derive the set $T = \{T_1, T_2, T_3\}$, $T' = \{T_1, T_2\}$, $T'' = \{T_3\}$. By Step 3, derive the outface subnets $PN_1$ of $T_1$ and $PN_2$ of $T_2$. By Step 4, for $T_1 \cap T_2 \neq \emptyset$, and $\exists p_\in PN_1$, $t_{14} \in T_3, p_{14} = p_\in PN_1$, then merge the outface subnet of $T_3$ into $PN_1$, thereby resulting in a workflow extended subnet $PN_1$. By the same steps, we merge the outface subnet of $T_3$ into $PN_2$ to obtain another workflow extended subnet $PN_2$. By Step 5, delete $t^*$ and its corresponding arcs $(o, t^*)$ and $(t^*, i)$, we obtain two workflow subnets $PN_1$ and $PN_2$ as shown in Figs.3 and 4.

Fig. 3. The decomposition subnet $PN_1$ of $PN$.

Fig. 4. The decomposition subnet $PN_2$ of $PN$.

In the decomposition algorithm, solving linear equations $A^TX = 0$ with the constraints $X > 0$ can be verified in polynomial time$^{[21,28]}$ in most instances. But the complexity is $O(m^n)$ at the worst case, where $n = |T|$, $m = |P|$. The processes of Steps 3 and 4 perform the construction and merging of their outface subnets in $PN$, and requires polynomial time. At the worst case, every transition associates with $m$ places in $PN$. Thus, the complexity is $O(m^n)$. Steps 1 and 5 are trivial. Therefore, the complexity of the decomposition algorithm is $O(m^n)$. Li et al.$^{[6]}$ provides a decomposition algorithm, and Ref.$^{[23]}$ extends it. Both algorithms deal with only workflow models or WF-nets with free-choice, and the complexity of these algorithms is also exponential, and their application is thus limited. Compared with Ref.$^{[6]}$ and Ref.$^{[23]}$, our algorithm only requires that the WF-nets be sound and well-structured, and reduces the maximum possible number of enumeration.

V. Related Work

Petri net decomposition is one of the most attractive techniques for supporting hierarchical Petri net models and reducing analysis complexity. There are many decomposition methods used in the simplification of model analysis. Zaitsev$^{[24]}$ defines functional subnets that are the outface subnets generated by the subnet of places set, and an algorithm of polynomial complexity is constructed to decompose nets. It is our opinion that the method does not apply to system modeling, because it ignores the context semantic of a system. Using the idea of traversing nodes, Nishi$^{[25]}$ proposes a decomposition method under which transition firing sequences are traversed. A timed Petri Net model for multiple entities can be decomposed into several subnets in which the optimal firing sequence for each subnet is easily solved in polynomial time. It is applied to a flow-shop scheduling problem. Li et al.$^{[6]}$ propose an algorithm to decompose a free-choice and acyclic Petri nets into a set of T-components. They improve the algorithm to decompose a free-choice and acyclic Petri nets into a set of T-components$^{[23]}$. Both methods are used to perform the workflow-net analysis, but the algorithms are also of exponential complexity.

VI. Conclusions

Compared with the existing algorithm, the proposed algorithm overcome the shortage of the methods based on depth traversal algorithm. This paper also proposes a new method to verify the soundness of a workflow net. In the future research, we will derive performance results of workflow system by combining the proposed decomposition algorithm and Stochastic Petri net, and apply this algorithm in command and control systems$^{[29,30]}$ and dynamic service composition$^{[31,32]}$.

References


PANG Shanchen is an associate professor of Computer Science at the College of Information Science and Engineering in Shandong University of Science and Technology. He is a member of the Chinese Professional Committee of Computer Society of Petri nets. He received Ph.D. degree in computer science from Tongji University in 2008. He is currently engaged in post-doctoral research at Tsinghua University.

His current main research interests include Petri net theory and application, formal methods, distributed concurrent system analysis and verification. He is currently host a China National Natural Science Foundation Project, as an academic backbone to participate in a National Key Basic Research Program Project (973 Program). He has won the second prize and the third prize of Shandong scientific research institutions of higher learning. (E-mail: shanchen-pang2008@gmail.com).

LIU Chuang was born in 1948, Ph.D., professor, Ph.D. supervisor. His current research interests include computer networks, performance evaluation, network security, Petri net theory, trustworthy networks and trustworthy computing. (E-mail: chlin@tsinghua.edu.cn).

ZHOU Mengchu is a professor of the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, USA. His research interests are in computer-integrated systems, Petri nets, semiconductor manufacturing, multi-lifecycle engineering, and system security. (E-mail: zhou@njit.edu).

LI Yun received the B.S. degree in computer science and technology from Beijing University of Posts and Communications, Beijing, China in 2009. He is currently a Ph.D. student in computer science at Tsinghua University. His research interests include Petri nets, workflow, service computing, cloud computing, QoS of networks.