PAPER *Special Section on Formal Approaches*

Computational Complexity and Polynomial Time Procedure of Response Property Problem in Workflow Nets

Muhammad Syafiq BIN AB MALEK†a), ***Nonmember***, **Mohd Anuaruddin BIN AHMADON**†b), ***Member***, ***and*** **Shingo YAMAGUCHI**†c), ***Senior Member***

SUMMARY Response property is a kind of liveness property. Response property problem is defined as follows: Given two activities α and β, whenever α is executed, is β always executed after that? In this paper, we tackled the problem in terms of Workflow Petri nets (WF-nets for short). Our results are (i) the response property problem for acyclic WF-nets is decidable, (ii) the problem is intractable for acyclic asymmetric choice (AC) WF-nets, and (iii) the problem for acyclic bridge-less well-structured WF-nets is solvable in polynomial time. We illustrated the usefulness of the procedure with an application example.
key words: *the response property, Petri net, process tree, computational complexity, polynomial time algorithm*

1. Introduction

Response property is a kind of liveness property. The response property problem is defined as follows: Given two activities α and β, whenever α is executed, is β always executed after that? Response property analysis is important to verify the correctness of a workflow process. As a simple example, Fig. 1 shows a Petri net [1] representing a workflow for an online ordering process. There are two cases in this workflow. In case 1, if the order is less than \$2000 (*Price less than* \$2000), then the ID of the buyer will be checked by the system. Next, the buyer can proceed to order (*Make Order*). Finally, the system will accept the order (*Instant Acceptance*). In case 2, if the order is more than \$2000 (*Price more than* \$2000), then the buyer must proceed directly to order the item (*Make Order*) and the manager will manually accept the order (*Manager Acceptance*). In this case, whenever *Price more than* \$2000 is executed, is *Manager Acceptance* always executed? The answer is no. This is because even if the price is above \$2000 (*Price more than* \$2000), then the system can accept the order (*Instant Acceptance*) instead of the manager. In this situation, the business rule is violated. Hence, it is important to distinguish the execution of each task to verify whether the workflow follows the business rule or not.

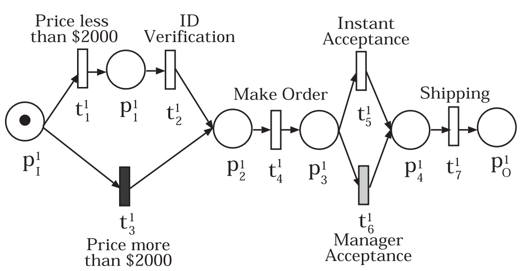


Fig. 1 A workflow N_1 of an online ordering service.

In Ref. [2], Hichami et al. proposed a method to verify response property based on Petri nets. The feature of the method is to introduce abstraction technique in order to reduce the computation time of model checking. However, the method lacks formality and did not take account of two or more firing of transition.

In this paper, we take an analytical approach to the response property problem. We first formalize the response property problem and then reveal the computational complexity of the problem. Then, we propose a polynomial time procedure to solve the problem with a representational bias of Petri net called as process tree. This paper is organized as follows: After the introduction in Sect. 1, Sect. 2 gives the formal definition and properties of Petri nets. In Sect. 3, we give the formal definition of the response property problem in terms of Petri nets. In Sect. 4, we reveal its decidability and computational complexity. Then, in Sect. 5, we give a polynomial time procedure to solve the problem and show an application example of our approach in Sect. 6. Finally, we give the conclusion and the future work.

2. Preliminaries

(1) Petri nets and workflow nets

A Petri net is a three tuple N=(P, T, A), where P, T, and A ($\subseteq(P\times T)\cup(T\times P)$) are finite sets of places, transitions, and arcs, respectively. Let x be a node of N. $\overset{N}{\bullet}x$ and $x\overset{N}{\bullet}$ respectively denote $\{y|(y, x)\in A\}$ and $\{y|(x, y)\in A\}$. A marking (or a state) is a mapping M: $P\rightarrow\{0, 1, 2, \cdots\}$. We represent M as a bag over P: M=$[p^{M(p)}|p\in P, M(p)>0]$. A transition t is said to be firable in M if $M\geq\overset{N}{\bullet}t$. Firing t in M results in a new marking M' (=$M\cup t\overset{N}{\bullet} - \overset{N}{\bullet}t$). This is denoted by $M[N, t\rangle M'$. A marking M_n is said to be reachable from a marking M_0 if there exists a transition firing sequence σ=$t_1t_2\cdots t_n$ such that $M_0[N, t_1\rangle M_1[N, t_2\rangle M_2\cdots[N, t_n\rangle M_n$. M_0 denotes the ini-

Manuscript received August 25, 2017.
Manuscript revised December 30, 2017.
Manuscript publicized March 16, 2018.
†The authors are with the Graduate School of Science and Technology for Innovation, Yamaguchi University, Ube-shi, 755–8611 Japan.
a) E-mail: w092vg@yamaguchi-u.ac.jp
b) E-mail: anuar@yamaguchi-u.ac.jp
c) E-mail: shingo@yamaguchi-u.ac.jp
DOI: 10.1587/transinf.2017FOP0008

tial marking in N. The set of all possible transition firing sequences from M_0 in N is denoted by $L(N, M_0)$. The set of all markings reachable from M_0 in (N, M_0) is denoted by $R(N, M_0)$. The tree representation of the markings in $R(N, M_0)$ is called the reachability tree [3].

N is said to be a workflow net (WF-net for short) [4] if (i) N has a single source place p_I and a single sink place p_O and (ii) every node is on a path from p_I to p_O. Each transition represents an action. For any WF-net, M_0 is given as $[p_I]$. We make N strongly connected by connecting p_O to p_I via an additional transition t^*. The resulting Petri net is called the *short-circuited net* of N, and is denoted by $\overline{N}$ $(=(P, T\cup\{t^*\}, A\cup\{(p_O, t^*), (t^*, p_I)\}))$. There are four important subclasses of WF-nets used in this paper: *well-structured* (WS for short), *extended free choice* (EFC for short), *asymmetric choice* (AC for short), and *process tree based* (PTB for short). N is said to be WS if there are neither disjoint paths from a place to a transition nor disjoint paths from a transition to a place in $\overline{N}$. N is said to be EFC if $\forall p_1, p_2 \in P : p_1\overset{N}{\bullet}\cap p_2\overset{N}{\bullet}\neq\emptyset \Rightarrow p_1\overset{N}{\bullet}=p_2\overset{N}{\bullet}$. N is said to be AC if $\forall p_1, p_2 \in P : p_1\overset{N}{\bullet}\cap p_2\overset{N}{\bullet}\neq\emptyset \Rightarrow p_1\overset{N}{\bullet}\subseteq p_2\overset{N}{\bullet}$ or $p_2\overset{N}{\bullet}\subseteq p_1\overset{N}{\bullet}$. A WF-net is a marked graph (MG for short) iff $\forall p \in P$, $|\overset{N}{\bullet}p|=|p\overset{N}{\bullet}|=1$.

Soundness is a criterion of correctness for WF-nets. A WF-net N is said to be sound iff (i) $\forall M\in R(N, [p_I])$: $\exists M'\in R(N, M)$:$M'\geq[p_O]$; (ii) $\forall M\in R(N, [p_I])$:$M\geq[p_O]\Rightarrow M=[p_O]$ and (iii) There is no dead transition in $(N, [p_I])$. The soundness of EFC WF-nets or WS WF-nets can be solved in polynomial time [4].

(2) Process Tree and Process Tree Based WF-net

A process tree is a tree representation of a process [5]. A process is represented by an action which is defined by an atomic action [6]. Each leaf node and each internal node respectively represents an action and an operator in the process.

Definition 1: [7] The set Π of process trees π is as follows:

(i) If α is an action, then $\alpha \in \Pi$.
(ii) If $\oplus$ is an operator and $\alpha_1, \alpha_2, \cdots, \alpha_n$ are actions, then $\oplus(\alpha_1, \alpha_2, \cdots, \alpha_n) \in \Pi$.
(iii) If $\oplus$ is an operator and $\pi_1, \pi_2, \cdots, \pi_n \in \Pi$, then $\oplus(\pi_1, \pi_2, \cdots, \pi_n) \in \Pi$. □

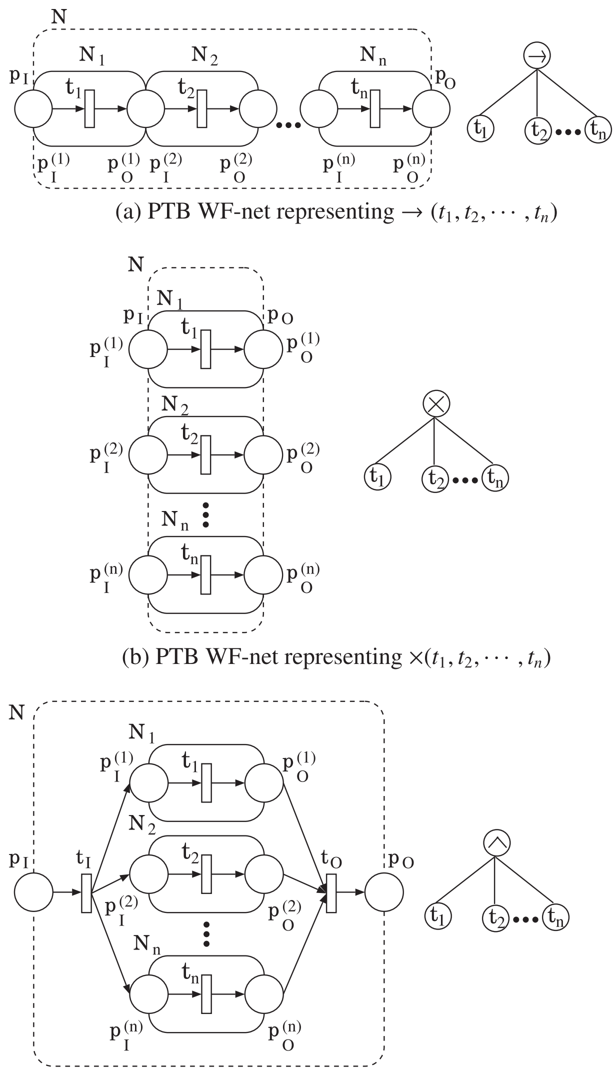
We use three operators standardized by the Workflow Management Coalition (WfMC for short) [4], [8]: sequence ($\rightarrow$), exclusive-choice ($\times$), and parallel ($\wedge$). Each operator can be translated to a part of a WF-net (See Fig. 2).

Next, we define a WF-net that can be converted into process tree called as Process Tree-based WF-net (PTB WF-net for short) [7], [9]. In PTB WF-net, each transition represents only unique action.

Definition 2 (PTB WF-net [7]): For any process tree π, let N be the WF-net itself and N_i $(i=1, 2, \cdots, n)$ be the subnet in N.

(i) If π is an action, a WF-net N which consists of a transition representing the action and its input and output places is PTB.
(ii) If π is $\oplus(\pi_1, \pi_2, \cdots, \pi_n)$, then let $N_1, N_2, \cdots, N_n$ be respectively PTB WF-nets representing sub-process trees $\pi_1, \pi_2, \cdots, \pi_n$.

a. If $\oplus$ is sequence, then a WF-net constructed by concatenating $N_1, N_2, \cdots, N_n$ which connects the sink place of N_i with the source place of $N_{i+1} (1\leq i<n)$ is PTB.
b. If $\oplus$ is exclusive choice, then a WF-net constructed by bundling PTB WF-nets $N_1, N_2, \cdots, N_n$ which forms a selection of concurrent paths between their source places and sink places is PTB.
c. If $\oplus$ is parallel, then a WF-net constructed by joining respectively all source places with a transition t_I, and sink places with a transition t_O of PTB WF-net $N_1, N_2, \cdots, N_n$ is PTB. □



(a) PTB WF-net representing $\rightarrow(t_1, t_2, \cdots, t_n)$

(b) PTB WF-net representing $\times(t_1, t_2, \cdots, t_n)$

(c) PTB WF-net representing $\wedge(t_1, t_2, \cdots, t_n)$

Fig. 2 Illustration of PTB WF-net and its equivalent process tree.

Noted that a process tree of PTB WF-net is an ordered tree. Soundness and well-structuredness are necessary condition to represent process tree [9].

3. Response Property

Response property is a kind of liveness property that given two transitions t and u, if t is fired, u has to be eventually fired after that. We restricted the analysis of response property to acyclic WF-net. This restriction is reasonable because acyclic WF-nets are applicable to analyze most actual workflows.

Let $\mathcal{L}(N, [p_I])$ ($\subseteq L(N, [p_I])$) be the set of any firing sequence that transforms $[p_I]$ to any dead marking. Note that in any acyclic WF-net N=(P, T, A), each marking in $R(N, [p_I])$ eventually reaches a dead marking.

Definition 3 (Response Property): For a transition pair t and u in an acyclic WF-net N, u is said to respond to t if $\forall\sigma \in \mathcal{L}(N, [p_I]) : \forall i \in \{1, 2, \cdots, |\sigma|\}^{\dagger}$: $(\sigma\{i\}=t \Rightarrow \exists j \in \{(i+1), (i+2), \ldots, |\sigma|\} : \sigma\{j\}=u)$. □

Definition 4 (Response Property Problem):
Instance: Acyclic WF-net N, Transitions t and u of N.
Question: Does u respond to t? □

Let us consider two instances of the response property problem as examples. The first instance is shown in Fig. 1. In this instance, every transition fires at most once. The second instance is shown in Fig. 3. In this instance, some of transitions fire twice or more. The instance in Fig. 1 has been considered in Sect. 1. We concretely discuss the reason in this section.

Instance 1:
Instance: Acyclic WF-net $(N_1, [p_I])$, transitions t^1_3 and t^1_6 (See Fig. 1).
Question: Does t^1_6 respond to t^1_3? □

The answer for Instance 1 is no, because t^1_6 does not always fire for $[p^1_I][\sigma, N_1\rangle[p^1_O]$ because there exist firing sequences $t^1_1 t^1_2 t^1_4 t^1_5$ or $t^1_3 t^1_4 t^1_5$ transforming $[p^1_I]$ to $[p^1_O]$. Existence of place branching at p^1_4 allows the t^1_6 to be skipped as t^1_5 is fired. Then t^1_6 does not respond to t^1_3. This means that Petri net structure plays an important role to the response property and the reachability of marking is important to the analysis of response property as discussed in Sect. 1.

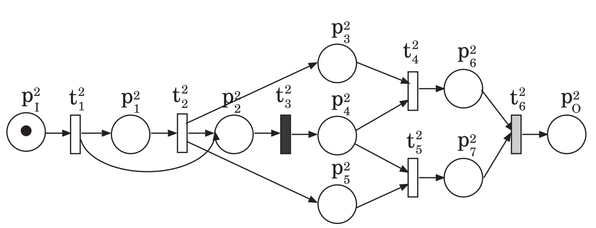


Fig. 3 Instance 2: WF-net N_2.

$^{\dagger}\{i, (i+1), \cdots, j\}$ denotes the set of integers from i to j.

Instance 2:
Instance: Acyclic WF-net $(N_2, [p^2_I])$, transitions t^2_3 and t^2_6 (See Fig. 3).
Question: Does t^2_6 respond to t^2_3? □

The answer for Instance 2 is yes. t^2_6 fires once for $[p^2_I][\sigma, N_2\rangle[p^2_6, p^2_7]$ when there are firing sequences $t^2_1 t^2_2 t^2_3 t^2_4 \mathbf{t^2_3} t^2_5 \mathbf{t^2_6}$ or $t^2_1 t^2_2 t^2_3 t^2_5 \mathbf{t^2_3} t^2_4 \mathbf{t^2_6}$ in $\mathcal{L}(N_2, [p_I])$. The value of i of t^2_3 is 5 which is lower than the value of j of t^2_6 that is 7 for both firing sequences. Then, transition t^2_6 is always fires after transition t^2_3 fires. Concretely, the order of transition firing plays an important role in deciding the response property.

As a transition t can fire many times, it is important to check each firing of transition. However, we can simplify the checking process with the property below:

Property 1: For a transition pair t and u in an acyclic WF-net N, u is said to respond to t iff $\forall\sigma \in \mathcal{L}(N, [p_I]) : (t \in \sigma$ and let i_{max} denote the last position of t in $\sigma \Rightarrow \exists j \in \{(i_{max}+1), (i_{max}+2), \cdots, |\sigma|\} : \sigma\{j\}=u)$. □

Proof : Let σ be any firing sequence in $\mathcal{L}(N, [p_I])$. The "only-if" part is obvious. We have only to prove the "if" part. We divide the proof into two cases where $t \notin \sigma$ and $t \in \sigma$. In the case of $t \notin \sigma$, the "if" part of the definition of the response property is obviously true. In the other case ($t \in \sigma$), t appears in σ more than once. Let k be any position of t in σ. $k \leqq i_{max}$ is obvious. t at position k is followed by the last one. From the assumption ($\exists j \in \{(i_{max}+1), (i_{max}+2), \cdots, |\sigma|\} : \sigma\{j\}=u$), the last t is followed by u. From the transitive law, we have that t at any position is followed by u. **Q.E.D.**

4. Computational Complexity and Intractability

Let us consider the decidability of the response property problem.

Theorem 1: The response property problem is decidable for acyclic WF-nets. □

Proof : Let N be any acyclic WF-net. Since N is acyclic, $\mathcal{L}(N, [p_I])$ is obviously finite. To check whether a transition α responds to another transition β in N, we have only to check the precedence relation between α and β for each sequence σ of $\mathcal{L}(N, [p_I])$: That is, if σ includes α, then β must appear at a later position than the last α. **Q.E.D.**

Let us consider the computational complexity of the response property problem for a subclass of acyclic WF-nets, called acyclic AC WF-nets.

We show that an NP-complete problem, called 3-conjunctive normal form boolean satisfiability problem (3-CNF-SAT for short), can be transformed to the response property problem of acyclic AC WF-nets.

Definition 5 (3-CNF-SAT):
Instance: Expression $\mathcal{E}$ of 3-conjunctive normal form that has n boolean variables and m clauses.
Question: Is there an assignment of variables satisfying

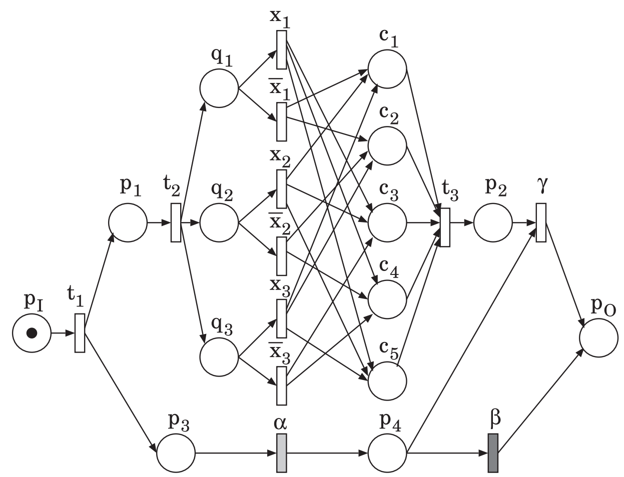


Fig. 4 The acyclic AC WF-net $(N_{\mathcal{E}_1},[p_1])$ corresponding to a 3-CNF-SAT expression $\mathcal{E}_1 = (\overline{x_1}\vee x_2\vee x_3) \wedge (\overline{x_1}\vee\overline{x_2}\vee x_3) \wedge (x_1\vee x_2\vee\overline{x_3}) \wedge (x_1\vee\overline{x_2}\vee\overline{x_3}) \wedge (x_1\vee x_2\vee x_3)$. $\mathcal{E}_1$ is satisfiable. β does not respond to α in $(N_{\mathcal{E}},[p_I])$

$\mathcal{E}$=*true*? □

Let us consider an acyclic AC WF-nets shown in Fig. 4. We need to check does β respond to α? The constructed WF-net shows that if $\mathcal{E}$=*true* then marking $[p_2, p_4]$ is reachable from $[p_I]$. However, β will not always fire because of the conflict with γ. In this case, β does not respond to α. We give the following theorem on the complexity:

Theorem 2: The response property problem is co-NP hard for acyclic AC WF-nets. □

Proof : We prove the NP-hardness by a reduction from 3-CNF-SAT. Let $\mathcal{E}$ be an expression of 3-CNF-SAT which has n boolean variables $x_1, x_2, \cdots, x_n$ and m clauses $c_1, c_2, \cdots, c_m$. A literal ℓ_i is either a variable x_i or its negation $\overline{x_i}$. Without loss of generality, it can be assumed that $\mathcal{E}$ has all of x_i's and $\overline{x_i}$'s $(1{\le}i{\le}n)$, and $m{\ge}3$.

We construct a WF-net $N_{\mathcal{E}}=(P_{\mathcal{E}}, T_{\mathcal{E}}, A_{\mathcal{E}})$ with two transitions α and β, and show that $\mathcal{E}$ is satisfiable iff β does not respond to α in $(N_{\mathcal{E}}, [p_I])$. $N_{\mathcal{E}}=(P_{\mathcal{E}}, T_{\mathcal{E}}, A_{\mathcal{E}})$ is given as follows.

$$
\begin{aligned}
P_{\mathcal{E}} &= \{p_I, p_1, p_2, p_3, p_4, p_O\} \cup \textstyle\bigcup_{i=1}^{n}\{q_i\} \cup \bigcup_{j=1}^{m}\{c_j\}\\
T_{\mathcal{E}} &= \{t_1, t_2, t_3, \alpha, \beta, \gamma\} \cup \textstyle\bigcup_{i=1}^{n}\{x_i, \overline{x_i}\}\\
A_{\mathcal{E}} &= \{(p_I, t_1), (t_1, p_1), (p_1, t_2), (t_3, p_2), (p_2, \gamma), (t_1, p_3),\\
&\quad (p_3, \alpha), (\alpha, p_4), (p_4, \gamma), (\gamma, p_O), (p_4, \beta), (\beta, p_O)\}\\
&\quad \cup \textstyle\bigcup_{i=1}^{n}\{(t_2, q_i), (q_i, x_i), (q_i, \overline{x_i})\}\\
&\quad \cup \textstyle\bigcup_{k=1}^{3}\bigcup_{j=1}^{m}\{(\ell_k, c_j) | \ell_k \text{ is the } k\text{-th literal of clause } c_j\}\\
&\quad \cup \textstyle\bigcup_{j=1}^{m}\{(c_j, t_3)\}
\end{aligned}
$$

$N_{\mathcal{E}_1}$ is an AC WF-net because places p_2 and p_4 share an output transition γ while p_4 has another output transition β; Places $c_1, c_2, \cdots, c_m$ share only one output transition t_2, and the other places share no output transition. $N_{\mathcal{E}}$ can be constructed in polynomial time, because it consists of $(n{+}m{+}6)$ places, $(2n{+}6)$ transitions, and $(3n{+}4m{+}12)$ arcs.

The proof of "if" part: Let μ denote an assignment of variables satisfying $\mathcal{E}$=*true*, and let $\ell_1, \ell_2, \cdots, \ell_n$ be the literals mapped to *true* by μ. By the construction of $N_{\mathcal{E}_1}$, we have

$$
\begin{aligned}
&[p_1]\ [N_{\mathcal{E}}, t_1t_2\rangle\ [q_1, q_2, \cdots, q_n, p_3]\\
&\qquad [N_{\mathcal{E}}, \ell_1\ell_2\cdots\ell_n\rangle\ M.
\end{aligned}
$$

Since μ satisfies $\mathcal{E}$, for each clause c_j $(1{\le}j{\le}m)$, there exists a literal ℓ_i $(1{\le}i{\le}n)$ in c_j. Therefore place c_j is marked by firing ℓ_i i.e. $M \geqq [c_1, c_2, \cdots, c_m, p_3]$. Let $M' = M - [c_1, c_2, \cdots, c_m, p_3]$.

$$
\begin{aligned}
M &= M' \cup [c_1, c_2, \cdots, c_m, p_3]\\
&\quad [N_{\mathcal{E}}, t_3\rangle M' \cup [p_2, p_3]\\
&\quad [N_{\mathcal{E}}, \alpha\gamma\rangle M' \cup [p_O]
\end{aligned}
$$

β is dead in $(N_{\mathcal{E}}, M' \cup [p_O])$. Thus, β does not respond to α.

The proof of "only-if" part: Let μ denote any assignment of variables satisfying $\mathcal{E}$=*false*. Since μ does not satisfy $\mathcal{E}$, there exists a clause c_j $(\in\{c_1, c_2, \cdots, c_m\})$ mapped to *false* by μ. Let $\ell_1^j,\ \ell_2^j,\ \ell_3^j$ denote the literals in c_j. Since the corresponding transitions $\ell_1^j,\ \ell_2^j,\ \ell_3^j$ do not fire, their common output place, i.e. place c_j, is never marked. c_j is an input place of transition t_3, so t_3 is dead. We have

$$
\begin{aligned}
&[p_I]\ [N_{\mathcal{E}}, t_1\rangle\ [p_1, p_3]\\
&\qquad [N_{\mathcal{E}}, *\rangle\ [p_3] \cup M \qquad (\forall M \in R(N_{\mathcal{E}}, [p_1]))\\
&\qquad [N_{\mathcal{E}}, \alpha\rangle\ [p_4] \cup M\\
&\qquad [N_{\mathcal{E}}, *\rangle\ [p_4] \cup M' \qquad (\forall M' \in R(N_{\mathcal{E}}, M))\\
&\qquad [N_{\mathcal{E}}, \beta\rangle\ [p_O] \cup M'
\end{aligned}
$$

Thus β responds to α. **Q.E.D.**

This means that the original problem is intractable for acyclic AC WF-nets. For example, let us consider the following boolean expression:

$$
\begin{aligned}
\mathcal{E}_1 &= (\overline{x_1}\vee x_2\vee x_3) \wedge (\overline{x_1}\vee\overline{x_2}\vee x_3) \wedge (x_1\vee x_2\vee\overline{x_3})\\
&\quad \wedge (x_1\vee\overline{x_2}\vee\overline{x_3}) \wedge (x_1\vee x_2\vee x_3)
\end{aligned}
$$

$\mathcal{E}_1$ is satisfiable by choosing x_1=*true*, x_2=*true*, x_3=*true*. Figure 4 shows the Petri net $N_{\mathcal{E}_1}$ constructed from $\mathcal{E}_1$. β does not respond to α, because

$$
\begin{aligned}
&[p_I]\ [N_{\mathcal{E}_1}, t_1t_2\rangle\ [p_1, p_3]\\
&\qquad [N_{\mathcal{E}_1}, x_1x_2x_3\rangle\ [{c_1}^2, {c_2}^1, {c_3}^2, c_4, {c_5}^3, p_3]\\
&\qquad [N_{\mathcal{E}_1}, t_3\rangle\ [c_1, c_3, {c_5}^2, p_2, p_3]\\
&\qquad [N_{\mathcal{E}_1}, \alpha\rangle\ [c_1, c_3, {c_5}^2, p_2, p_4]\\
&\qquad [N_{\mathcal{E}_1}, \gamma\rangle\ [c_1, c_3, {c_5}^2, p_O].
\end{aligned}
$$

5. Polynomial-Time Verification of Response Property

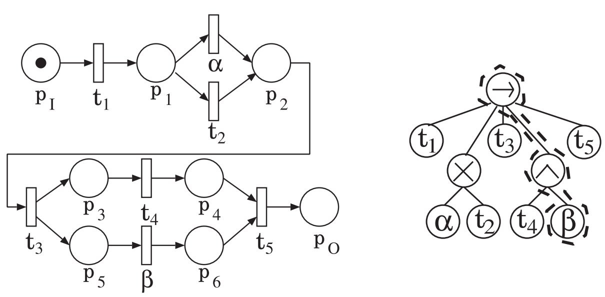
In this section, we propose a polynomial time procedure to decide response property.

5.1 Structural Analysis for Response Property Problem

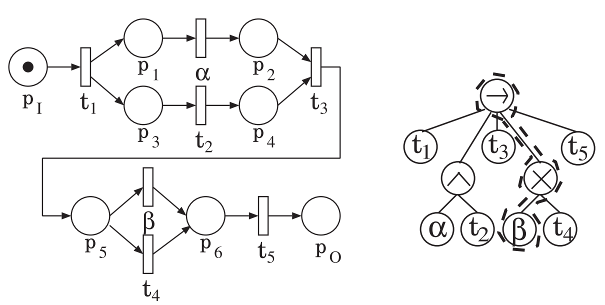
A transition with two or more output places is called a transition-split, and a transition with two or more input places is called a transition-join. Similarly, a place with two or more output transitions is called a place-split, and a place with two or more input transitions is called a place-join [10].

Acyclic bridge-less WS WF-net has balance structure of transition/place-splits with transition/place-joins. Two disjoint paths initiated by a transition-split are joined by a place-join. Two disjoint paths initiated by a place-split are joined by a transition-join. The structure of a transition-split with a transition-join is called the transition split-join and the structure of a place-split with a place-join is called the place split-join. Acyclic bridge-less WS WF-nets have nesting structure of transition split-join structure and place split-join structure [10]. This enables us to check the response property not in the node-level but in those split-join structure-level.

Let us consider two instances of acyclic bridge-less WS WF-net shown in Fig. 5 to solve the response property problem. Both instances have one transition split-join structure and one place split-join structure. In N_1, β responds to α. α appeared in the disjoint path between place-split p_1 and place-join p_2. β appeared in the disjoint path between transition-split t_3 and transition-join t_5. Since β is included within a transition split-join structure, β always fire after α. In N_2, β does not respond to α. α appeared in the disjoint path between transition-split t_1 and transition-join t_3. β appeared in the disjoint path between place-split p_5 and place-join p_6. Place-join p_2 allows β to be skipped when t_4 is fired. The result of the response property is changed by the combination of transition/place split-join structure and the position of α and β. Therefore, we need to consider the combination of disjoint path and position of transitions within the WF-net.



(a) Acyclic bridge-less WS WF-net N_1. β responds to α.



(b) Acyclic bridge-less WS WF-net N_2. β does not respond to α.

Fig. 5 Examples of acyclic bridge-less WS WF-nets.

5.2 Utilization of Process Tree

Acyclic bridge-less WS WF-net is known as PTB WF-net [9]. Figure 6 shows the general form of process tree of PTB WF-net. We can represent transition split-join as parallel operator ($\wedge$). We can also represent place split-join as exclusive-choice operator ($\times$). Let υ_{NCA} be the nearest common ancestor node of α and β in a process tree π of a WF-net N where there exists the shortest path from α via υ_{NCA} to β (See Fig. 6).

First, we consider two cases when υ_{NCA} is an exclusive-choice operator ($\times$) or a parallel operator ($\wedge$). If υ_{NCA} is an exclusive-choice operator ($\times$), β does not respond to α. We can give the following Lemma:

Lemma 1: In a PTB WF-net, if υ_{NCA}=$\times$, t_j does not respond to t_i (See Fig. 2 (b)). □

Proof : See Appendix A.1. **Q.E.D.**

If υ_{NCA} is a parallel operator ($\wedge$), β does not respond to α. We can give the following Lemma:

Lemma 2: In a PTB WF-net, if υ_{NCA}=$\wedge$, t_j does not respond to t_i (See Fig. 2 (c)). □

Proof : See Appendix A.2. **Q.E.D.**

Next, if υ_{NCA} is a sequence operator ($\rightarrow$) and the parent of α and β, β responds to α. However, if υ_{NCA} is a sequence operator ($\rightarrow$) and not the parent of α and β, β does not always respond to α. Also, if the position of α is on the left of β in π_N, then β responds to α because α will fire before β in the sequence construct in N. However, if α is on the right of β, then β does not respond to α because β will fire before α.

Let us consider again the two instances of acyclic bridge-less WS WF-net shown in Fig. 5. We converted both instances into process tree (See the right side of Fig. 5). Let us focus on the dotted line on both process tree. For process tree of N_1, we find that if parallel operator ($\wedge$) exists between υ_{NCA} and β, then β responds to α. Otherwise, for process tree of N_2, we find that if exclusive-choice operator ($\times$) exists between υ_{NCA} and β, then β does not respond to α. We can give the following theorem:

Theorem 3: For a PTB WF-net N with transitions α and β, β responds to α iff in the process tree Π_N

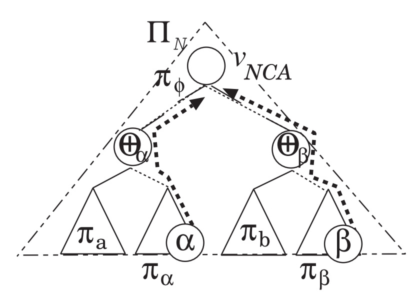


Fig. 6 General form of process tree for a PTB WF-net. υ_{NCA} denotes the nearest common ancestor node, $\oplus_\alpha$ denotes the ancestor node of α and $\oplus_\beta$ denotes the ancestor node of β which are the descendants of υ_{NCA}.

(i) The nearest common ancestor v_{NCA} of α and β is a sequence operator ($\rightarrow$);
(ii) The position of α is on the left side of $\beta^\dagger$; and
(iii) There is no exclusive-choice operator ($\times$) on the path between v_{NCA} and β. □

Proof : The proof of the "if" part: Conditions (i), (ii) and (iii) can be illustrated by the process tree Π_N shown in Fig. 6. $\oplus_\alpha$ denotes the ancestor nodes of α and the descendants of v_{NCA}. $\oplus_\beta$ is defined similarly as $\oplus_\alpha$. Π_N can be illustrated with the WF-net N shown in Fig. 7. N consists of N_A, N_B, and N_C. If v_{NCA} is a non-parent node, then α and β is in a different connected subnets in N. α is in N_A, β is in N_B, and N_C connects N_A and N_B. t_I of N_B can always fire after α and β can always be enabled after t_I since N_B has no choice. WF-net N is sound. Therefore, $\overline{N}$ can be regarded as an interconnection of MG components and from the property of MG, there exists a path connecting α and β. Hence, β responds to α in N.

The proof of the "only-if" part: We show the proof using a process tree Π_N which has a subtree π_ϕ with root v_{NCA} where α and β are the leaf nodes.

In Condition (i), if v_{NCA}=$\times$, based on Lemma 1, the firing is selective where only either transition in the subtree π_α or π_β will fire. If v_{NCA}=$\wedge$, based on Lemma 2, there exist partial firing sequences $\alpha\sigma\beta$ and $\beta\sigma\alpha$ where there is a case when β can fire before α.

In Condition (ii), if α is on the right of β then i>j. Therefore, there exists a case where β fires before α in N. N is acyclic, therefore the firing of each transitions is only once where there exists no firing sequence that satisfies $\forall i_{max} \in \{1, 2, \cdots, |\sigma|\} : \sigma\{i_{max}\}=\alpha \Rightarrow \exists j \in \{(i_{max}+1), (i_{max}+2), \cdots, |\sigma|\} : \sigma\{j\}=\beta$ once β fires before α.

In Condition (iii), if there exists an exclusive-choice operator between the path from v_{NCA} to β such that $\oplus_\beta$=$\times$, then β is in an exclusive-choice construct. Therefore, there exists a case where β will not fire. **Q.E.D.**

Theorem 3 implies the condition to decide the response property problem. So we construct a polynomial time procedure based on Theorem 3. The process tree can be traversed with Breadth-first Search (BFS) [11]. We give the procedure as follows:

≪Decision of Response Property≫
Input: PTB WF-net N (=(P, T, A)), transitions $\alpha, \beta \in T$.
Output: Does β respond to α?

1° Convert N into process tree Π with ≪Process Tree Conversion Algorithm≫ [9].
2° Check Conditions (i), (ii) and (iii) of Theorem 3.
  2-1° ▹ Check Condition (i).
  Let the ρ_α be the path from root to α and ρ_β be the path from root to β. Let ρ_C be the common part of ρ_α and ρ_β. If the last node of ρ_C is not sequence operator ($\rightarrow$), then output no and stop.
  2-2° ▹ Check Condition (ii).
  Let i be the position of α and j be the position of β. If i>j, then output no and stop.
  2-3° ▹ Check Condition (iii).
  Backtrack from β to v_{NCA}. If a visited node is exclusive-choice operator ($\times$), then output no and stop.
3° Output yes and stop.

Property 2: The following problem can be solved in polynomial time: Given a PTB WF-net N with two transitions α and β, to decide whether β responds to α. □

Proof : The ≪Process Tree Conversion Algorithm≫ takes $O(|P| + |T|)$ and response property check takes $O(|T|)$ based on BFS. **Q.E.D.**

6. Application Example

In this section, we show an application example by showing the response property analysis of a secure online shopping process (see Fig. 1). We assume that we need to check the responsiveness between these two important activities of the online shopping process which is "*Price more than* \$2000" and "*Manager Acceptance*". Let us verify the instance using our procedure.

In Step 1°, we convert N_1 to process tree Π_{N_1} (See Fig. 8). In Step 2°, we check the Condition (i), (ii) and (iii) of Theorem 3. In Step 2-1°, we check the nearest common node v_{NCA} of α and β. We obtained that ρ_α=$t_3^1 \times \rightarrow$, ρ_β=$t_6^1 \times \rightarrow$, and ρ_C= $\rightarrow$, hence v_{NCA}= $\rightarrow$. In Step 2-2°, we check Condition (ii). From i=3 and j=6, we obtained that

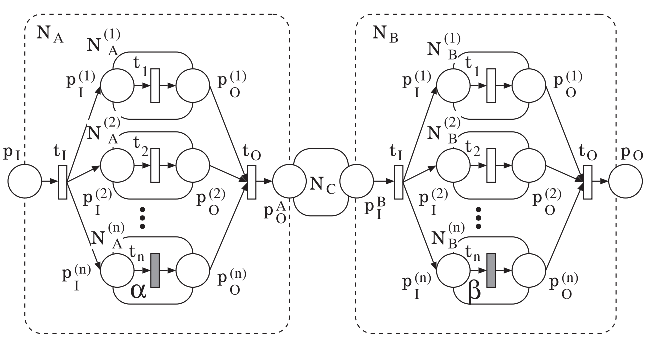


Fig. 7 Illustration of the proof of "if" part of Theorem 3.

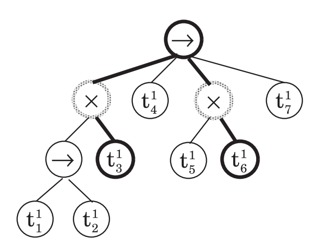


Fig. 8 Process tree Π_{N_1} representing N_1. The bold area shows that t_6^1 does not respond to t_3^1. The grey area shows an exclusive choice part which will cause restriction of nodes to be selected.

†Noted that a process tree is an ordered tree.

$i < j$. t^1_3 is on the left side of t^1_6. Then, in Step 2-3° we check Condition (iii). We backtrack from t^1_6 to v_{NCA} and found $\times \rightarrow$ on the path. There is a $\times$ on the path. In N_1, there is a place p^1_3 which constructs exclusive-choice operator ($\times$). Either t^1_5 or t^1_6 will be fired. So, t_6 does not always fire. Finally, in Step 3°, the procedure outputs no. Thus, we obtained that t^1_6 does not respond to t^1_3.

7. Conclusion

In this paper, we gave the formal definition of the response property problem. We also showed that the problem is decidable. We also revealed the intractability for AC WF-nets. We constructed a polynomial time procedure for PTB WF-nets based on the given theorem. Response property analysis is important in terms of analyzing the execution of an action whenever another action is executed in order to adapt to specifications in systems and business processes. In our future work, we plan to investigate the response property problem for cyclic WF-nets.

Acknowledgements

This research was partially supported by Interface Corporation, Japan and Malaysia-Japan Higher Education Program.

References

[1] T. Murata, "Petri nets: Properties, analysis and applications," Proc. IEEE, vol.77, no.4, pp.541–580, 1989.
[2] O.E. Hichami, M.A. Achhab, I. Berrada, R. Oucheikh, B.E.E. Mohajir, "An approach of optimization and formal verification of workflow Petri nets," Journal of Theoretical and Applied Information Technology, vol.61, no.3, pp.486–495, 2014.
[3] S. Yamaguchi, "Polynomial time verification of reachability in sound extended free-choice workflow net," IEICE Trans. Fundamentals, vol.E97-A, no.2, pp.468–475, Feb. 2014.
[4] W.M.P. van der Aalst and K.M. van Hee, Workflow Management: Models, Methods, and Systems, The MIT Press, 2002.
[5] W. van der Aalst, J. Buijs, and B. van Dongen, "Towards improving the representational bias of process mining," Data-Driven Process Discovery and Analysis, Lecture Notes in Business Information Processing, vol.116, pp.39–54, Springer-Verlag, Berlin, 2012.
[6] R. De Nicola, "Process algebras," Encyclopedia of Parallel Computing, pp.1624–1636, 2011.
[7] M.A.B. Ahmadon and S. Yamaguchi, "State number calculation problem of workflow nets," IEICE Trans. Inf. & Syst., vol.E98-D, no.6, pp.1128–1136, June 2015.
[8] W.M.P. van der Aalst, "Verification of workflow nets," Application and Theory of Petri Nets 1997, Lecture Notes in Computer Science, vol.1248, pp.407–426, Springer, 1997.
[9] M.A.B, Ahmadon and S. Yamaguchi, "Convertibility and conversion algorithm of well-structured workflow net to process tree," Proc. CANDAR 2013, pp.122–127, 2013.
[10] N. Qu, S. Yamaguchi, and Q.-W. Ge, "Parallel degree of well-structured workflow nets," IEICE Trans. Fundamentals, vol.E93-A, no.12, pp.2730–2739, Dec. 2010.
[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, Second Edition, pp.998–1003, The MIT Press, 2001.

Appendix

A.1 Proof of Lemma 1 : If v_{NCA}=$\times$, then the PTB WF-net N is constructed by bundling $N_1, N_2, \cdots, N_n$ which forms a selective concurrent paths between their source places and sink places. In N_n, p_I=$p^{(1)}_I$=$p^{(2)}_I$=$\cdots$=$p^{(n)}_I$ and p_O=$p^{(1)}_O$=$p^{(2)}_O$=$\cdots$=$p^{(n)}_O$. Note that only one path will allow the transition t_n in N_n to fire. Therefore there exists different firing sequences $\sigma_1, \sigma_2, \cdots, \sigma_n$ for each path in $N_1, N_2, \cdots, N_n$ for $[p_I][N_n, \sigma_n\rangle[p_O]$. Hence, only either t_i or t_j will fire. **Q.E.D.**

A.2 Proof of Lemma 2 : If v_{NCA}=$\times$, then the PTB WF-net N is constructed by joining respectively all source places of the concurrent paths with a transition t_I, and sink places with a transition t_O of PTB WF-net $N_1, N_2, \cdots, N_n$. Since $N_1, N_2, \cdots$, and N_n are sound, $[p^{(1)}_O, p^{(2)}_O, \cdots, p^{(n)}_O]$ is reachable from $[p^{(1)}_I, p^{(2)}_I, \cdots, p^{(n)}_I]$. $p^{(1)}_O, p^{(2)}_O, \cdots$, and $p^{(n)}_O$ are connected to p_O via another additional transition t_O. Therefore there exist partial firing sequences σ=$t_i\sigma_1 t_j$ and σ=$t_j\sigma_1 t_i$ where σ_1 is all possible firing sequences of transitions in T for $[p^{(1)}_I, p^{(2)}_I, \cdots, p^{(n)}_I][N_n, \sigma\rangle[p^{(1)}_O, p^{(2)}_O, \cdots, p^{(n)}_O]$. Hence, based on σ, t_j will not always fire after t_i. **Q.E.D.**

Muhammad Syafiq Bin Ab Malek received his B.E. degree from Yamaguchi University, Japan in 2016. He is a graduate student in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan. His research interest includes Petri net and its application.

Mohd Anuaruddin Bin Ahmadon received his B.E., M.E. and D.E. degrees from Yamaguchi University, Japan in 2014, 2015 and 2017, respectively. He is currently an Assistant Professor in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan. His research interest includes Petri net and its application to system analysis. He is a member of IEEE.

Shingo Yamaguchi received the B.E., M.E. and D.E. degrees from Yamaguchi University, Japan, in 1992, 1994 and 2002, respectively. He was a Visiting Scholar in the Department of Computer Science at University of Illinois at Chicago, United States, in 2007. He is currently a Professor in the Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan. His research interests are in the area of net theory and its applications. He is a senior member of IPSJ and IEEE.