Analyzing Non-determinism in Telecommunication Services Using P-Invariant of Petri-Net Model

Masahide Nakamura, Yoshiaki Kakuda and Tohru Kikuno
Department of Information and Computer Sciences
Faculty of Engineering Science, Osaka University
Machikaneyama 1-3, Toyonaka City, Osaka 560, Japan
E-mail: {masa-n, kakuda, kikuno}@ics.es.osaka-u.ac.jp

Abstract
The non-deterministic behaviors in telecommunication services are well-known as one of the most typical Feature Interactions, and they should be detected and eliminated from the telecommunication service specifications. The conventional analysis method of this non-determinism is based on reachability analysis. Since the method must exhaustively enumerate all reachable global states, it cannot be applied to the complex communication services which include many users.

In this paper, we propose an alternative method based on a Petri-net model. The method constructs a logically equivalent Petri-net for a given service specification, and obtains a set of states which cause the non-deterministic behaviors using rules in the service specification. Then, the method identifies states in the set which are not reachable from the initial state using P-invariant of the Petri-net, and deletes them from the set. As P-invariant is utilized as the necessary condition, we must finally apply reachability analysis to states in the resultant set. Since the number of states in the resultant set may be reduced to relatively small, the new method enables us to analyze the more complex services.

1 Introduction
The future advanced networks, such as Intelligent Networks (IN)[11], will drastically increase the requirement and development of new telecommunication services. In developing such new services, the services have to be designed so that they satisfy some fundamental and desirable properties(e.g., all of services are free from deadlocks). Moreover, designers have to check if the new services conflict with features of the existing services or not. Recently, this conflict checking becomes the essential problem for development of new telecommunication services, and the conflict is generally called feature interaction[2].

With respect to feature interaction, one of the most desirable properties in telecommunication services is that "any service must be provided deterministically in the system". Telecommunication services can be often modeled by a state transition machine, in which a state consisting of local states of service users successively moves to a next state by a trigger of user's event. If multiple transitions are allowed to be executed for a certain pair of a state and a user's event, then a non-deterministic transition occurs and this non-deterministic transition may cause an illegal state change against the user's intention. This kind of non-deterministic behavior is also well-known as one of the most typical feature interaction [3, 4, 9, 10], which should be detected and eliminated.

The conventional analysis method is based on reachability analysis. This method at first enumerates all possible reachable states by applying reachability analysis to the state transition machine, and then checks the existence of non-determinism for each reachable state[5]. However, it takes a lot of time and space, because the number of reachable states exponentially increases along with the increase of the number of users. So it may be impossible for the design process of complex services, which include many users and user's events, to apply this conventional analysis.

In this paper, we propose an alternative new analysis method which doesn't require state enumeration at the first stage. In our approach, we at first construct a logically equivalent Petri-net for a given service specification, then obtain a set of all the states which may cause the non-deterministic behavior using rules in the service specification. Then we identify all states in the set which are not reachable from the initial state using P-invariant of the Petri-net, and delete them from the set. As P-invariant is used as necessary condition, we must finally apply reachability analysis to states in the resultant set. By using the proposed method, we can reduce the number of states in the resultant set drastically. As a result, the proposed method may reduce the cost of analyzing non-determinism in a given telecommunication service specification, and enable us to design the more complex services.

2 Practical Example
Example 1 Let us consider a service which has both Call Waiting (CW) feature and Call Forwarding Variable (CFV) feature [11, 12]. CW feature provides such a capability that a CW user can receive an additional call from a third party when the CW user is talking with someone. On the other hand, CFV feature forwards the incoming call to the terminal number preset by the CFV user. Suppose that user A subscribes
both features CW and CFV whose forwarding address is user D, and that A is talking with user B. In this situation, if user C makes a call to user A, then should the call from C be received by A or forwarded to D?

Many other examples of non-deterministic behaviors are presented in [3, 9, 10](e.g., combination service of CW and Three Way Calling(TWC) feature, etc.)

3 Preliminaries

3.1 Service Specification

Service specification studied in this paper is defined as a set of rules of service logic such as STR[7] and declarative transition rule[4]. These rule-based methods have been widely studied towards the practical use since (a) the modularity of the rule facilitates the addition or modification of the new service, and (b) a simple IF-THEN form of each rule enables non-experts to easily design the service logic[9].

Definition 1 A service specification S is defined by $S = (R, s_0)$, where $R$ is a finite set of rules and $s_0$ is an initial global state(or simply initial state). Each rule $r \in R$ is defined as follows.

$$r : A_1, \ldots, A_p \xrightarrow{e} B_1, \ldots, B_t$$

$A_i$ or $B_j$ is called a predicate and is represented by $p(x_1, \ldots, x_k)$, where $p$ represents a predicate symbol and $x_1, \ldots, x_k$ are variables. $e$ is called a user's event (or simply event) and is represented by $e(x_1, \ldots, x_k)$, where $e$ represents an event symbol. Next, we interpret that $A_1, \ldots, A_p$ and $B_1, \ldots, B_t$ represent AND-conjunctions of predicates $A_i$'s and $B_j$'s, and we call them pre-condition and post-condition of rule $r$, respectively. The pre-condition is allowed to include negation of predicate such as $\neg p(x_1, \ldots, x_k)$. For convenience, let $e[r]$ denote the event of rule $r$ in the following.

A global state(or simply state) is an AND-conjunction of all instances of predicates in rules, and is represented by

$$C_1, \ldots, C_q$$

where each $C_i$ is an instance of predicate defined by $p(a_1, \ldots, a_k)$, where $a_1, \ldots, a_k$ are constants which identify the service users. For convenience, in the following we list up only the instances which take true value at the state and may refer it also as the state. In particular, the initial state $s_0$ is generally defined as

$$s_0 = idle(U_1), idle(U_2), \ldots, idle(U_n)$$

where $U_i$ is an identifier for specifying a service user and $n$ is the number of the users.

Definition 2 A state can be changed to the next state by application of a rule. Let $s$ be a current state of the service and let $r$ denote an instantiation of a rule $r \in R$ based on a substitution $\theta$. If state $s$ includes the pre-condition of $r\theta$, then we say rule $r$ is applicable to $s$ for $\theta$ and we rewrite the corresponding predicates in $s$ into the post-condition of $r\theta$. As the result, a new state $s'$ is generated from $s$, which is interpreted as “current state $s$ moves to next state $s'$ by a trigger of event in $r\theta$”. A state $s$ is reachable from an initial state $s_0$ if there exists at least one sequence of states such that $s_0, s_1, \ldots, s_j = s$, where each $s_i (1 \leq i \leq j)$ is the next state of $s_{i-1}$.

Example 2 The following is an example of a rule which describes a fundamental function “Suppose that $x$ receives a dial-tone and $y$ is idle. In this situation, if $x$ dials $y$, then $x$ will be calling $y$’ of telephone service:

$$r : dialtone(x), idle(y) \xrightarrow{dial(x, y)} calling(x, y)$$

In rule $r$, the pre-condition of $r$ is “dialtone($x$), idle($y$)”, the post-condition of $r$ is “calling($x$, $y$)”, and the event of rule $r$ is $e[r] = dial(x, y)$. Next, we show an example of a state $s$ which represents that users A,B and D are idle and user C receives a dialtone.

$$s = idle(A), idle(B), dialtone(C), idle(D)$$

If we apply a substitution $\theta = \{x|C, y|D\}$ to rule $r$, then we obtain

$$r\theta : dialtone(C), idle(D) \xrightarrow{dial(C, D)} calling(C, D)$$

Since state $s$ includes the pre-condition of $r\theta$, rule $r$ is applicable to $s$ for $\theta$. As a result, the event dial($C$, $D$) changes the state into a next state $s'$ defined by

$$s' = idle(A), idle(B), calling(C, D)$$

Example 3 The following shows an example of service specification of simplified Plain Ordinary Telephone Service (POTS). For simplicity, we assume that the number of users is only four ($A, B, C, D$).

$$R = \{$$

$$r_1 : idle(x) \xrightarrow{offhook(x)} dialtone(x)$$

$$r_2 : dialtone(x) \xrightarrow{onhook(x)} idle(x)$$

$$r_3 : dialtone(x), idle(y) \xrightarrow{dial(x, y)} busytone(x)$$

$$r_4 : dialtone(x), idle(y) \xrightarrow{dial(x, y)} busytone(x)$$

$$r_5 : calling(x, y) \xrightarrow{onhook(x)} idle(x), idle(y)$$

$$r_6 : calling(x, y) \xrightarrow{offhook(y)} talk(x, y)$$

$$r_7 : talk(x, y) \xrightarrow{onhook(y)} idle(x), busytone(y)$$

$$r_8 : talk(x, y) \xrightarrow{offhook(y)} idle(x), busytone(y)$$

$$r_9 : busytone(x) \xrightarrow{onhook(x)} idle(x)$$

$$s_0 = idle(A), idle(B), idle(C), idle(D)$$

3.2 Non-deterministic Behaviors

Intuitively, the non-deterministic behavior occurs when two or more rules are simultaneously applicable to a global state for the identical instance of event. The non-deterministic behavior on a global state $s$ is formally defined as follows.
Definition 3 Let $S = (R, s_0)$ be a service specification. Then, we say that "state $s$ causes a non-deterministic behavior" iff $s$ satisfies both of the following conditions.

Condition P1: $s$ is reachable from $s_0$.
Condition P2: There exist at least two different rules $r_1, r_2 \in R$ and two substitutions $\theta_1, \theta_2$ such that $r_1$ and $r_2$ are applicable to $s$ for $\theta_1$ and $\theta_2$, respectively, and that $e[r_1\theta_1] = e[r_2\theta_2]$ holds.

Now, we explain the non-deterministic behaviors using Example 1, again.

Example 4 Consider the following two rules $r_1$ and $r_2$ and state $s$.

\[ r_1 : \begin{array}{ll} & CW(x), talk(x, y), dialtone(z) \\
& \rightarrow \begin{array}{l} \text{dialtone}(z) \\
\end{array} \\
\end{array} \\
\]

\[ r_2 : \begin{array}{ll} & CFV(y, z), idle(z), dialtone(x) \\
& \rightarrow \begin{array}{l}
\text{dialtone}(x), \text{calling}(z, z) \\
\end{array} \\
\end{array} \\
\]

\[ s = CW(A), CFV(A, D), talk(A, B), dialtone(C), idle(D) \]

Rule $r_1$ implies a CW feature such that "A CW user $x$ can receive an additional call from a third party $z$ while $x$ is talking with $y$". And rule $r_2$ implies a CFV feature such that "If a CFV user $y$ sets the forwarding address to $z$, the call to $y$ is forwarded to $z$". State $s$ means that user A has both CW feature and CFV feature with forwarding to D, D is idle. Now, we suppose that $s$ is reachable. It is clear that $r_1$ is applicable to state $s$ for $\theta_1 = \{z\rightarrow A, y\rightarrow B, z\rightarrow C\}$, and $e[r_1\theta_1] = \text{dial}(C, A)$. Simultaneously, $r_2$ is also applicable to $s$ for $\theta_2 = \{z\rightarrow C, y\rightarrow A, z\rightarrow D\}$, and $e[r_2\theta_2] = \text{dial}(C, A)$. Thus, $e[r_1\theta_1] = e[r_2\theta_2]$, and Condition P2 holds. So, $s$ causes the non-deterministic behavior. This is exactly the one explained in Example 1.

4 Previous Analysis Method

The goal of the analysis discussed in this paper is to check if there exist such states that satisfy both conditions P1 and P2 in Definition 3 for the given service specification $S$. Here, let $U, S_1$, and $S_2$ denote a set of all global states, a set of states satisfying Condition P1 and a set of states satisfying Condition P2, respectively. The goal of the analysis is to determine the intersection of $S_1$ and $S_2$.

Phase 1: Enumerate all reachable states from $s_0$ by exhaustive applications of rules.

Phase 2: Check Condition P2 for each reachable state obtained in Phase 1.

Figure 1 shows a schematic representation of the conventional approach. At first, Phase 1 identifies $S_1$ (i.e., a set of reachable state) and then Phase 2 extracts $S_1 \cap S_2$ by applying Condition P2 to each state in $S_1$.

Figure 1: Concept of conventional method

Let us briefly estimate the time complexity of this approach. In the following, $m$ and $n$ denote the number of rules and the number of users in the service specification, respectively.

Cost of Phase 1 ($C_1$): This depends on the number of the reachable states. As an instance, consider the service specification of the simplified POTS in Example 3. If the number of users who take part in the service is four, then the number of reachable states is 345. In the cases of 5, 6 and 7 users, the number of reachable states exponentially grows to 2043, 13029 and 88119, respectively. From this observation, it is natural to estimate $C_1$ to be exponential order of $n$.

Cost of Phase 2 ($C_2$): To check whether a state $s$ satisfies Condition P2 or not, we have to apply each pair of rules with the same event symbol to the state $s$. The number of those pairs is bounded by $m(m - 1)/2$. So, $C_2$ is estimated as $m(m - 1)/2 * C_1 \approx m^2 * C_1$.

Total Cost: Since $C_1$ is exponential order of $n$, the total cost is also exponential of $n$. Thus, the previous method needs a lot of time and cost (especially if we apply it to the services in which many users take part).

In this paper, we propose an alternative analysis method using a high level Petri-Net. Next section presents a kind of high level Petri-Net model onto which the service specification is mapped.

5 New Petri-Net Model

5.1 Labeled Pr/T Net

In this section, we define a kind of Petri-Net which is an extension of predicate transition nets (Pr/T Nts)[8].

Definition 4 A labeled Pr/T net $N$ is defined by $N = (P, T, F, U, E, L, L_I, M_0)$, where

(1) $P$ is a set of places.
(2) $T$ is a set of transitions, and $P \cap T = \phi$.
(3) $F \subseteq (P \times T) \cup (T \times P)$ is flow relation. Each element of $F$ is called arc.
(4) $H \subseteq (P \times T)$ is a set of inhibitor arcs.
(5) $U$ is a set of constants.
(6) $V$ is a set of variables ranging over $U$.
(7) $E$ is a set of predicate, and each element is represented by $e(x_1, ..., x_n)$, $x_i \in V$. 

10c.4.3 1255
(8) $L_a$ is the arc labeling function which attaches a label $(x_1, \ldots, x_k)$, where each $x_i \in V$, to the arc or the inhibitor arc, and $k$ is called arity of the arc. For any input/output arc of each place $p \in P$, its arity $k$ is a unique constant associated with $p$.

(9) $L_t$ is the transition labeling function which attaches the element of $E$ to each transition.

(10) $M_0$ is an initial marking (Marking will be defined in Definition 5).

In $t = \{ p(t, p) \in H \cup (F \cap (P \times T)) \}$ and Out $t = \{ p(t, p) \in F \cap (T \times P) \}$ are called input places and output places of transition $t$, respectively.

Remark 1 The differences between our labeled Pr/T net and Pr/T net are that (1) our model includes the inhibitor arcs to represent the negation of predicates (see Definition 1) and (2) the label is attached to each transition.

Definition 5 Color set of place $p$, denoted by $C(p)$, is the set of all constant k-tuples $(a_1, \ldots, a_k)$, where each $a_i \in U$ and $k$ is the arity of $p$. Each element of $C(p)$ is called colored token (or simply token) and can be allocated to a place $p$. The allocation of the tokens to each place $p \in P$ is called marking and it is defined as a mapping function from $P$ to the multiset over $C(p)$. A marking $M$ can be also expressed in terms of a vector: $M = (M(p_1), \ldots, M(p_m))$.

Let $Q(t)$ be a set of variables that occur at the incident arcs of $t$ and at the predicate on $t$. Let $x_1, \ldots, x_k$ be an arbitrary (but fixed) sequence of all variables in $Q(t)$. Then, color set of transition $t$, denoted by $C(t)$, is the set of all constant k-tuples $(a_1, \ldots, a_k)$ obtained by substituting each $x_i$ in the sequence by a constant in $U$. Thus, each color $c = (a_1, \ldots, a_k) \in C(t)$ can be interpreted as a substitution such that $(x_1 | a_1, \ldots, x_k | a_k)$. We represent this substitution by $\theta(c)$.

Definition 6 Consider $t \in T$, $c \in C(t)$, and a marking $M$. For $L_a(p, t)$, define $L_a(p, t) \theta(c)$ be a constant tuple obtained by substituting the variables in $L_a(p, t)$ according to $\theta(c)$. Then, $t$ is enabled for $\theta(c)$ under $M$ iff

$$\forall p \in \text{In}(t) \quad \{ L_a(p, t) \theta(c) \} \subseteq M(p)$$

$$\text{if } (p, t) \in F \quad \{ L_a(p, t) \theta(c) \} \notin M(p)$$

$$\text{if } (p, t) \in H$$

If $t \in T$ is enabled for $\theta(c)$ under $M$, then $t$ can fire. Firing of $t$ changes the current marking $M$ into the next marking $M'$ as follows:

$$M'(p) = \begin{cases} 
M(p) & \text{if } p \notin \text{In}(t) \cup \text{Out}(t) \\
M(p) \cup_m \{ L_a(p, t) \theta(c) \} & \text{if } p \in \text{In}(t) \ \text{and } \text{Out}(t) \\
M(p) \cup_m \{ L_a(p, t) \theta(c) \} & \text{if } p \in \text{Out}(t) \ \text{and } \text{In}(t) \\
M(p) - m_a \{ L_a(p, t) \theta(c) \} & \text{if } p \in \text{In}(t) \ \text{and } \text{Out}(t) \\
M(p) - m_a \{ L_a(p, t) \theta(c) \} & \text{if } p \in \text{Out}(t) \ \text{and } \text{In}(t) \\
M(p) & \text{if } p \in \text{In}(t) \ \text{and } \text{Out}(t)
\end{cases}$$

where $\cup_m$ and $-m_a$ are the union and difference operations defined on multisets [8].

Figure 2: An explanation of firing

A marking $M$ is called reachable from $M_0$ iff $M = M_0$ or there exists at least one sequence of marking $M_0, M_1, \ldots, M_n = M$ such that $M_{i+1}$ is a next marking of $M_i$.

Example 5 We explain the firing of transitions using Figure 2. Consider the marking $M$ in Figure 2(a), which is also specified by

\begin{align*}
\text{idle} & \hspace{1cm} \text{diatone} \hspace{1cm} \text{calling} \\
M & = (\{(A), (B), (D)\}, \{(C)\}, \phi) 
\end{align*}

For example, take transition $t_2$ and $\theta(C, D) = (z_1 | C, y_1 | D)$. Then $\text{In}(t_2) = \{ \text{diatone}, \text{idle} \}$, and

$$L_a(\text{diatone}, t_2) \theta(C, D) = \{ (C) \}$$

and $L_a(\text{idle}, t_2) \theta(C, D) = \{ (D) \} \subseteq M(\text{idle})$. Thus, $t_2$ is enabled for $\theta(C, D)$ under $M$.

Now, suppose that $t_2$ fires for $\theta(C, D)$ under $M$. Then, tokens $\langle C \rangle$ and $\langle D \rangle$ are respectively removed from places diatone and idle, since $\{ \text{diatone}, \text{idle} \}$ and $\{ \text{diatone}, \text{idle} \} = \{ \langle C \rangle \} \subseteq M(\text{idle})$. Moreover, a new token $\langle C, D \rangle$ is allocated to place calling, because $\{ \text{calling} \} = \{ \langle C, D \rangle \} \subseteq M(\text{calling})$. As the result, $M$ is transformed into the following next marking $M'$, which is also shown in Figure 2(b).

\begin{align*}
\text{idle} & \hspace{1cm} \text{diatone} \hspace{1cm} \text{calling} \\
M' & = (\{(A), (B)\}, \phi, \{(C), (D)\}) 
\end{align*}

5.2 Service Specification Net

Here, we define the particular labeled Pr/T net for a service specification $S$.

Definition 7 Let $S = (R, s_0)$ be a service specification. Then, a service specification net $N(S) = (P, T, F, H, U, V, E, L_a, L_t, M_0)$ for a given service specification $S$ is a labeled Pr/T net which satisfies the following conditions.

(1) $E$ is a set of all events in rules of $S$.

(2) $P$ is a set of all predicate symbols in rules of $S$. 

(3) $U$ is a set of all service users in $s_0$.
(4) $V$ is a set of all variables in rules of $S$.
(5) For each rule $r_i \in R$, there is exactly one transition $t_i \in T$ such that $L(t_i) = [r_i]$.
(6) For each predicate $p_{ij}(x_{i1},...,x_{im})$ in precondition of rule $r_i \in R$, exactly one arc with a label $(x_{i1},...,x_{im})$ exists from place $p_{ij}$ to transition $t_i$.
(7) For each predicate $\neg p_{ij}(x_{i1},...,x_{im})$ in precondition of rule $r_i \in R$, exactly one inhibitor arc with a label $(x_{i1},...,x_{im})$ exists from transition $t_i$ to place $p_{ij}$.
(8) For each predicate $p_{ij}(x_{i1},...,x_{im})$ in postcondition of rule $r_i \in R$, exactly one arc with a label $(x_{i1},...,x_{im})$ exists from transition $t_i$ to place $p_{ij}$.
(9) If the initial state $s_0$ includes $p(c_1,\ldots,c_m)$, then the initial marking $M_0(p) = \{c_1,\ldots,c_m\}$.

According to Definition 7, we can easily understand that (1) the pre(post)-condition of a rule corresponds to the input(output) places of a transition, (2) the event of a rule corresponds to the predicate attached to a transition, (3) the initial state corresponds to the initial marking.

Remark 2 A state $s$ of $S$ uniquely corresponds to a marking $M$ on $N(S)$. That is, if a predicate $p(a_1,\ldots,a_n)$ holds (that is, takes the true value) on state $s$, then place $p$ has a token $(a_1,\ldots,a_n)$ under $M$. Suppose that states $s$ and $s'$ correspond to markings $M$ and $M'$, respectively, and that rule $r_i$ corresponds to transition $t_i$. Then, a state transition from $s$ to $s'$ by rule $r_i$ exactly corresponds to a firing of transition $t_i$ which transforms $M$ into $M'$.

Example 6 Consider again Example 5. Then a labeled Petri net shown in Figure 2(a) is a service specification net for the service specification consisting of the following two rules.

\[ r_1 : \text{idle}(x) \xrightarrow{\text{offhook}(x)} \text{dialtone}(x) \]
\[ r_2 : \text{dialtone}(x),\text{idle}(y) \xrightarrow{\text{dial}(x,y)} \text{calling}(x,y) \]

Next, consider the following two states:

\[ s = \text{idle}(A),\text{idle}(B),\text{dialtone}(C),\text{idle}(D) \]
\[ s' = \text{idle}(A),\text{idle}(B),\text{calling}(C,D) \]

Then markings $M$ and $M'$ in Example 5 respectively represent these two states $s$ and $s'$. The state transition from $s$ to $s'$ by rule $r_2$ is already explained in Example 2. For this state transition, we can correspond it to a firing of $t_2$ for $\theta(C,D) = \{x|C,y|D\}$ which transforms $M$ into $M'$.

The following lemma implies that $N(S)$ is logically equivalent to $S$ with respect to reachability analysis.

Lemma 1 For a given service specification $S$ and a service specification net $N(S)$ for $S$, there exist one-to-one correspondence between a set of reachable markings of $N(S)$ and a set of reachable states of $S$.

Example 7 Figure 3 shows a service specification net obtained from the service specification of POTS in Example 3. We can completely simulate the behavior of service specification on this net model.

6 Proposed Analysis Method
6.1 Outline of Our Method
An outline of the proposed analysis method is shown in Figure 4. At first, we calculate a set $S_{P2}$ of states which satisfy Condition $P2$ (Phase 1) and then check reachability of each state in $S_{P2}$ (Phase 2 and Phase 3).

The set $S_{P2}$ can be easily calculated using the rules of service specification. Intuitively, a state in $S_{P2}$ can be generated by joining pre-conditions of two rules which have the same event symbol. Therefore, the essential problem to realize this new approach is how nicely we check the reachability of states in $S_{P2}$. On performing this checking, we utilize extensively $P$-invariant of the service specification net.

6.2 $P$-Invariant
Definition 8 [6] Let $N(S)$ be a service specification net with $u$ places and $v$ transitions, and let $p \in P$, $t \in T$ with $(p,t) \not\in H$. Then, $W(p,t)$ (or $W(t,p)$) is a linear function $[C(t) \rightarrow C(p)]$ such that $\forall c = (a_1,\ldots,a_i) \in C(t), W(p,t)(c) = T_{a}(p,t)\theta(c)$ (or,
Phase 3: Reachability Analysis

Figure 4: Outline of our method

W(t, p)(c) = \mathcal{L}_d(t, p)\mathcal{O}(c), respectively. In our discussion, we consider only four kinds of linear functions defined as follows: (1) identity function id, (2) zero function 0, (3) projection functions p1 and p2 such that p1(\alpha, \beta) = (\alpha), p2(\alpha, \beta) = (\beta).

Then the incident matrix of \mathcal{N}(S) is the u x v matrix defined as follows.

Then u-dimensional vector Y such that Y * A = 0 is called P-invariant of \mathcal{N}(S), where * is a formal product operation of matrix[6, 8].

Example 8 Consider the service specification net shown in Figure 3. Then the incident matrix A of this net is

\[
\begin{pmatrix}
t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \\
id & \text{id} & \text{id} & -p_2 & 0 & p_1 & 0 & p_1 & \text{id} \\
dil & \text{id} & -p_1 & -p_1 & 0 & 0 & 0 & 0 & 0 \\
cog & 0 & 0 & 0 & 0 & 0 & -\text{id} & -\text{id} & 0 & 0 & 0 \\
bst & 0 & 0 & p_1 & 0 & 0 & p_2 & p_1 & -\text{id} & 0 \\
ltk & 0 & 0 & 0 & 0 & 0 & 0 & -\text{id} & -\text{id} & -\text{id}
\end{pmatrix}
\]

The following vector Y is a P-invariant of \mathcal{N}(S) since a relation Y * A = 0 holds.

\[
\begin{pmatrix}
id \text{id} \text{id} \text{id} \text{id} \text{id} \text{id} \text{id} \text{id}
\end{pmatrix}
\]

We can apply the procedure for the calculation of the P-invariant[1, 6]. The following theorem for the P-invariant is a well-known theorem to be used for checking reachability.

Theorem 1 Let Y be the P-invariant of the service specification net \mathcal{N}(S). If marking M is reachable from the initial marking M_0, then Y * M^t = Y * M^t_0.

Remark 3 The equation Y * M^t = Y * M^t_0 is only a necessary condition for any reachable marking M. Hence, even if Y * M^t = Y * M^t_0 holds, we cannot conclude, in general, that M is reachable.

6.3 Analysis Algorithm \(\Omega\)

This subsection shows the analysis method of non-determinism. The input of the algorithm is a service specification \(\mathcal{S} = (R, \mathcal{S}_0)\). Figure 5 shows the proposed algorithm \(\Omega\). In the following, we briefly explain \(\Omega\).

In Phase 0, we first construct the service specification net \mathcal{N}(S), and then calculate P-invariant Y of \mathcal{N}(S) by applying the available method proposed in [1, 6]. MS_p is a set of markings each of which corresponds to a state in \(Sp_0\) (defined in Section 4).

Next, Phase 1 determines a set of states \(Sp_2\) satisfying Condition P2. At first, we construct a condition C' to which two rules r1 and r2 are simultaneously applicable(Step1-3). Then, we make a marking corresponding to C and extend it to all marking which covers C by wild-cards (they represents an arbitrary multisets of tokens)(Step4).

Next, in Phase 2 we check for each marking \(M \in MS_{p2}\), if M is not reachable from \(M_0\) by solving the equation \(Y * M^t = Y * M^t_0\). According to Theorem 1, if the equation is not solvable (i.e., there is no assignment of U to wild-cards which satisfy \(Y * M^t = Y * M^t_0\)), then M is not reachable. Hence, we delete M from \(MS_{p2}\). If the equation is solvable, then we do not derive any decision on the reachability of M, thus we leave M in \(MS_{p2}\).

Finally in Phase 3 we must apply conventional reachability analysis method to the resultant \(MS_{p2}\) (i.e., the size of \(MS_{p2}\) may be reduced using P-invariant in Phase 2). To be explained in Example 9 and in subsection 6.4, the cost needed in Phase 3 depends on the size of the resultant \(MS_{p2}\) in Phase 2.

Example 9 We apply the algorithm \(\Omega\) to the POTS specification in Example 4.

Phase 0(Preliminary): We obtain \(\mathcal{N}(S)\) shown in Figure 3. Then we calculate the following P-invariant:

\[
\begin{pmatrix}
id \text{id} \text{id} \text{id} \text{id} \text{id} \text{id} \text{id} \text{id}
\end{pmatrix}
\]

Phase 1(Decision of States in \(Sp_2\)):

Step1: We select the following rules with the same event symbol of offhook:

\[
\begin{align*}
r_1 : \text{idle} & \rightarrow \text{dialtone} \\
r_6 : \text{calling} & \rightarrow \text{busytone}
\end{align*}
\]

Step2: We can apply \(\theta_1 = \{x|A\}\) and \(\theta_6 = \{y|A\}\) to \(r_1\) and \(r_6\), respectively, since \(e[r_1\theta_1] = e[r_6\theta_6] = \text{offhook}(A)\). Then we get the following instances of rules \(r_1\) and \(r_6\):

\[
\begin{align*}
r_1\theta_1 : \text{idle}(A) & \rightarrow \text{dialtone}(A) \\
r_6\theta_6 : \text{calling}(x, A) & \rightarrow \text{busytone}(A)
\end{align*}
\]

Step3: By combining pre-conditions of \(r_1\theta_1\) and \(r_6\theta_6\), we get the following condition C:

\[
C = \text{idle}(A), \text{calling}(x, A)
\]
Analysis algorithm 0:

Phase 0 (Preliminary): Construct the service specification net \( N(S) \) for a given service specification \( S = (R, s_0) \). Then calculate \( P \)-invariant \( Y \) of \( N(S) \). Define a set \( MS_{p_0} \) to be a set of markings, each of which corresponds to a state in \( S_{p_0} \), and make the initial value of \( MS_{p_0} \) to be empty.

Phase 1 (Decision of States in \( S_{p_0} \)):

Step 1: Select two rules \( r_i \) and \( r_j \) from \( R \) whose event symbols are identical.

Step 2: Apply a pair of substitutions \( \theta_i \) and \( \theta_j \) such that \( e[\theta_i] = e[\theta_j] \) to \( r_i \) and \( r_j \), respectively.

Step 3: By combining two pre-conditions of \( r_i \theta_i \) and \( r_j \theta_j \), using AND operation, obtain a condition (say it condition \( C \)). If \( C \) forms null condition, we conclude that \( r_i \) and \( r_j \) are mutually exclusive with each other, and go to Step 5. Otherwise, go to Step 4.

Step 4: At first, put tokens to places based on predicates in \( C \). Then, to each place \( p \in P \), put the wild-card of tokens \( \Sigma(x_1, \ldots, x_k) \), where \( x_k \) is a variable and \( k \) is arity of \( p \). Construct a marking \( M_e \) which corresponds to the resulting net, and put \( M_e \) into \( MS_{p_2} \).

Step 5: If some pairs of rules to be checked still remain, then go to Step 1.

Phase 2 (Check of Unreachability using \( P \)-invariant): Check if for each marking \( M \in MS_{p_2} \), reachable by solving the equation \( Y \ast M = Y \ast M' \). If the equation is not solvable, then we conclude that state \( s \) corresponding to \( M \) is not reachable, and delete \( M \) from \( MS_{p_2} \).

Phase 3 (Reachability Analysis for Resultant \( MS_{p_2} \)): Apply so-called reachability analysis method to the resultant \( MS_{p_2} \) finally obtained in Phase 2.

Figure 5: Analysis algorithm \( \Omega \)

Step 4: At first, we obtain a marking \( M \) according to \( C \) as follows:

\[
\begin{align*}
\text{idle} & \quad \text{dialtone} & \quad \text{calling} & \quad \text{busytone} & \quad \text{talk} \\
M = & \quad (\{x_1\}, \emptyset, \{x, A\}, \emptyset, \emptyset)
\end{align*}
\]

Next, we put five wild-cards to places, and finally get the following marking \( M_e \):

\[
\begin{align*}
\text{idle} & \quad \text{dialtone} & \quad \text{calling} \\
M_e = & \quad (\{A\}, \Sigma(x_1), \{x, A\}, \Sigma(x_2), \Sigma(x_4), \Sigma(x_3), \Sigma(x_0), \Sigma(x_1))
\end{align*}
\]

In this example, we can get ten other markings in Phase 1, thus get \( |MS_{p_2}| = 11 \).

Phase 2 (Check of Unreachability using \( P \)-invariant): Consider marking \( M \) as an example. Then we get

\[
\begin{align*}
Y \ast M_0' & = \text{id}(\{A\}, \{B\}, \{C\}, \{D\}) + \text{id}(\emptyset) + (p_1 + p_2)(\emptyset) + \text{id}(\emptyset) + (p_1 + p_2)(\emptyset) \\
& = \{A\}, \{B\}, \{C\}, \{D\}.
\end{align*}
\]

\[
\begin{align*}
Y \ast M' & = \text{id}(\{A\}, \Sigma(x_1)) + \text{id}(\{\Sigma(x_2)\}) + (p_1 + p_2) \\
& \quad (\{x, A\}, \Sigma(x_3), x_4) + \text{id}(\Sigma(x_3)) + (p_1 + p_2)(\{\Sigma(x_0), x_7\})
\end{align*}
\]

For this, no matter how we nicely choose assignment of constant such as \( (A), (B), (C), (D) \) to \( x \) and \( \Sigma(x_1), \Sigma(x_0), \Sigma(x_1) \)’s, the equation \( Y \ast M' = Y \ast M_0' \) never holds (i.e., this equation is unsolvable). Therefore, we can conclude that \( M \) is not reachable.

Similarly, we check the reachability for other ten markings in \( MS_{p_2} \). As the result, all markings in \( MS_{p_2} \) are deleted and finally \( MS_{p_2} \) becomes an empty set in Phase 2. Therefore, in this example, Phase 3 is not executed. (We can guarantee that the service specification of POTS is free from non-deterministic behavior without any state enumeration in Phase 3.)

6.4 Properties of Our Method

Here, we discuss the correctness, the cost and the advantages of the proposed method.

Theorem 2 For any state \( s \), if \( s \) causes a non-deterministic behavior, then a marking \( M \) corresponding to \( s \) exists in \( MS_{p_2} \).

As shown in the algorithm \( \Omega \), we must apply Phase 3 (that is, conventional reachability analysis method) to the resultant set \( MS_{p_2} \). Thus, from Theorem 2 it is clear that algorithm \( \Omega \) (Phase 0, 1, 2 and 3) identifies \( S_{p_2} \cap S_{p_3} \).

Next, let us briefly estimate the cost of the algorithm \( \Omega \). In the following, \( m \) and \( n \) denote the number of rules and the number of users, respectively.
Cost of Phase 0 (C₀): The cost of transformation of a given service specification into the net model is obviously of order m. The calculation of P-invariant depends only on net structure and it needs the time of exponential order of m (thus, the cost doesn’t depend on n). So, C₀ = m + cₘ^n ≈ cₘ^m.

Cost of Phase 1 (C₁): This depends on the number of the states satisfying Condition P2 and the total number is normally exponential order of n. However, by utilizing the wild-card, the execution of Step 1 through Step 5 (i.e., the number of loops) in Phase 1 is bounded by the number of rule’s pairs with the same event symbol. So, C₁ is approximately estimated as m * (m - 1)/2 ≈ m².

Cost of Phase 2 (C₂): To check if a marking M is reachable or not, we have to evaluate Y * M² and Y * M. The number of total multiplications needed for these evaluations is equal to the dimension of P-invariant, i.e., the number of predicate symbols in all rules. This is generally bounded by polynomial order m. Finally, this checking must be repeated for all M’s in MS₀, thus C₂ = m² * C₁ = m² * m² ≈ m⁴.

Cost of Phase 3 (C₃): C₃ deeply depends on the size of resultant MS₂, and it is generally exponential order of n. Since MS₂ seems to depend on the service specification, C₃ cannot be evaluated reasonably without experimental application to many practical services (Surely if MS₂ = φ then C₃ = 0 as shown in Example 6).

Finally, the major advantages of our method are summarized as follows:
(a) By changing orders of phases in the conventional method, we can put the application of reachability analysis at the last phase of the algorithm O (rather than at the first phase in the conventional method). As the result, we may reduce the number of states drastically, for which the reachability analysis should be applied.
(b) By utilizing P-invariant for the reachability checking, we can execute state reduction very efficiently without any state enumeration.
(c) Especially for a case that a given service specification doesn’t include any non-determinism, we can very quickly convince it (at the end of Phase 2). The cost in this case is cₘ⁴ and thus it does not depend on the number of users.

7 Concluding Remarks
In this paper, we have proposed a new analysis method based on a Petri-net for checking non-determinism in a given service specification. The proposed method consists of four phases: construction of the service specification net (Phase 0), decision of non-determinism (Phase 1), checking of reachability (Phase 2) and reachability analysis (Phase 3). The most attractive point of our method is that the reduction of the cost in Phase 3 is realized efficiently using P-invariant in Phase 2.

As mentioned in Section 6, in order to show the usefulness of our method, we must execute the experimental evaluations using practical service specifications. Currently we are planning to apply the algorithm to practical services [11, 12], and are developing a computer aided tool for the experiments.

Acknowledgement
This research was partly supported by Kokusai Denshin Denwa Co., Ltd. (KDD). The authors would like to thank Prof. T. Ohta (Souka University), Mr. Y. Ueda (OKI Electric Industry Co., Ltd.), and research staffs in KDD R&D Laboratories for their helpful discussion.

References