

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 rules*[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 $S = \langle R, s_0 \rangle$, 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.

$$r : A_1, \dots, A_s \xrightarrow{E} B_1, \dots, B_t$$

A_i or B_j is called a *predicate* and is represented by $p(x_1, \dots, x_k)$, where p represents a predicate symbol and x_1, \dots, x_k are *variables*. E is called a *user's event* (or simply *event*) and is represented by $e(x_1, \dots, x_k)$, where e represents an event symbol. Next, we interpret that A_1, \dots, A_s and B_1, \dots, 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 $\neg p(x_1, \dots, x_k)$. 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

$$C_1, \dots, C_q$$

where each C_i is an *instance* of predicate defined by $p(a_1, \dots, a_k)$, where a_1, \dots, 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

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

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 $r\theta$ denote an instantiation of a rule $r \in R$ based on a substitution θ . If state s includes the pre-condition of $r\theta$, then we say rule r is *applicable to s for θ* 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 iff there exists at least one sequence of states such that $s_0, s_1, \dots, s_j = s$, 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 θ . 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).

$R = \{$

$$\begin{aligned} r_1 : & idle(x) \xrightarrow{offhook(x)} dialtone(x) \\ r_2 : & dialtone(x) \xrightarrow{onhook(x)} idle(x) \\ r_3 : & dialtone(x), \neg idle(y) \xrightarrow{dial(x,y)} busytone(x) \\ r_4 : & dialtone(x), idle(y) \xrightarrow{dial(x,y)} calling(x, y) \\ r_5 : & calling(x, y) \xrightarrow{onhook(x)} idle(x), idle(y) \\ r_6 : & calling(x, y) \xrightarrow{offhook(y)} talk(x, y) \\ r_7 : & talk(x, y) \xrightarrow{onhook(x)} idle(x), busytone(y) \\ r_8 : & talk(x, y) \xrightarrow{onhook(y)} idle(y), busytone(x) \\ r_9 : & busytone(x) \xrightarrow{onhook(x)} idle(x) \} \\ s_0 = & idle(A), idle(B), idle(C), idle(D) \end{aligned}$$

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 $\mathcal{S} = \langle R, s_0 \rangle$ 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 $r_i, r_j \in R$ and two substitutions θ_i, θ_j such that r_i and r_j are applicable to s for θ_i and θ_j , respectively, and that $\epsilon[r_i\theta_i] = \epsilon[r_j\theta_j]$ 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 .

$$\begin{aligned}
 r_1 : & \quad CW(x), talk(x, y), dialtone(z) \\
 & \quad \xrightarrow{dial(z, x)} CW(x), CWcalling(z, x), talk(x, y) \\
 r_2 : & \quad CFV(y, z), idle(z), dialtone(x) \\
 & \quad \xrightarrow{dial(x, y)} CFV(y, z), calling(x, z) \\
 s = & \quad CW(A), CFV(A, D), talk(A, B), dialtone(C), idle(D)
 \end{aligned}$$

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 set 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, A is talking with B, C receives a dialtone, and D is idle.

Now, we suppose that s is reachable. It is clear that r_1 is applicable to state s for $\theta_1 = \{x|A, y|B, z|C\}$, and that $\epsilon[r_1\theta_1] = dial(C, A)$. Simultaneously, r_2 is also applicable to s for $\theta_2 = \{x|C, y|A, z|D\}$, and $\epsilon[r_2\theta_2] = dial(C, A)$. Thus, $\epsilon[r_1\theta_1] = \epsilon[r_2\theta_2]$, 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 \mathcal{S} . Here, let U , S_{p_1} and S_{p_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_{p_1} and S_{p_2} .

The straight-forward and conventional analysis method[5] is based on the state enumeration, and consists of the following two phases.

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_{p_1} (i.e., a set of reachable state) and then Phase 2 extracts $S_{p_1} \cap S_{p_2}$ by applying Condition P2 to each state in S_{p_1} .

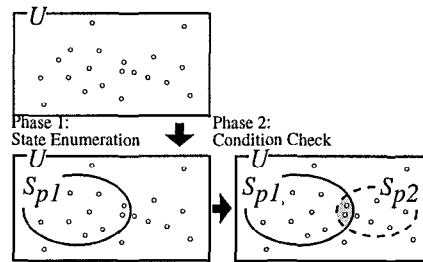


Figure 1: Concept of conventional method

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 Phase1 (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 Phase2 (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 Nets)*[8].

Definition 4 A labeled Pr/T net \mathcal{N} is defined by $\mathcal{N} = \langle P, T, F, H, U, V, E, L_a, L_t, M_0 \rangle$, 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 predicates, and each element is represented by $\epsilon(x_1, \dots, x_n)$, $x_i \in V$.

(8) L_a is the *arc labeling function* which attaches a label $\langle x_1, \dots, x_k \rangle$, 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).

$In(t) = \{p | (p, t) \in H \cup (F \cap (P \times T))\}$ 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 $\langle a_1, \dots, a_k \rangle$, 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 it 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: $M = (M(p_1), \dots, M(p_m))$.

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, \dots, x_l 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 l -tuples $\langle a_1, \dots, a_l \rangle$ obtained by substituting each x_i in the sequence by a constant in U . Thus, each color $c = \langle a_1, \dots, a_l \rangle \in C(t)$ can be interpreted as a substitution such that $\{x_1|a_1, \dots, x_l|a_l\}$. 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

$$\forall p \in In(t) \quad \begin{cases} \{L_a(p, t)\theta(c)\} \subseteq M(p) & \dots \text{ if } (p, t) \in F \\ \{L_a(p, t)\theta(c)\} \not\subseteq M(p) & \dots \text{ if } (p, t) \in H \end{cases}$$

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:

$$M'(p) = \begin{cases} M(p) & \dots \text{ if } p \notin In(t) \cup Out(t) \\ M(p) -_{ms} \{L_a(p, t)\theta(c)\} & \dots \text{ if } p \in In(t) - Out(t) \\ M(p) \cup_{ms} \{L_a(t, p)\theta(c)\} & \dots \text{ if } p \in Out(t) - In(t) \\ M(p) -_{ms} \{L_a(p, t)\theta(c)\} \cup_{ms} \{L_a(t, p)\theta(c)\} & \dots \text{ if } p \in In(t) \cap Out(t) \end{cases}$$

where \cup_{ms} and $-_{ms}$ are the union and difference operations defined on multisets[8].

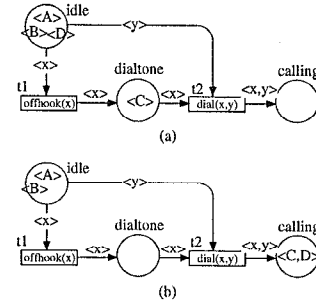


Figure 2: An explanation of firing

A marking M is called *reachable* from M_0 iff $M = M_0$ or there exists at least one sequence of marking $M_0, M_1, \dots, M_n = M$ 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

$$M = \begin{matrix} & \text{idle} & \text{dialtone} & \text{calling} \\ = & (\{A\}, \{B\}, \{D\}), & \{\{C\}\}, & \phi \end{matrix}$$

For example, take transition t_2 and $\theta(C, D) = \{x|C, y|D\}$. Then $In(t_2) = \{\text{dialtone}, \text{idle}\}$, and $\{L_a(\text{dialtone}, t_2)\theta(C, D)\} = \{\{C\}\} = M(\text{dialtone})$ 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 $\{C\}$ and $\{D\}$ are respectively removed from places *dialtone* and *idle*, since $\{\text{dialtone}, \text{idle}\} = In(t_2) - Out(t_2)$, $L_a(\text{dialtone}, t_2)\theta(C, D) = \{C\}$ and $L_a(\text{idle}, t_2)\theta(C, D) = \{D\}$. Moreover, a new token $\{C, D\}$ is allocated to place *calling*, because $\{\text{calling}\} = Out(t_2) - In(t_2)$ and $L_a(t_2, \text{calling})\theta(C, D) = \{C, D\}$. As the result, M is transformed into the following next marking M' , which is also shown in Figure 2(b).

$$M' = \begin{matrix} & \text{idle} & \text{dialtone} & \text{calling} \\ = & (\{A\}, \{B\}), & \phi, & \{\{C, D\}\} \end{matrix}$$

5.2 Service Specification Net

Here, we define the particular labeled Pr/T net for a service specification \mathcal{S} .

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

- (1) E is a set of all events in rules of \mathcal{S} .
- (2) P is a set of all predicate symbols in rules of \mathcal{S} .

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

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 \mathcal{S} uniquely corresponds to a marking M on $\mathcal{N}(\mathcal{S})$. That is, if a predicate $p(a_1, \dots, a_n)$ holds (that is, takes the true value) on state s , then place p has a token $\langle a_1, \dots, a_n \rangle$ 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 Pr/T net shown in Figure 2(a) is a service specification net for the service specification consisting of the following two rules.

$$\begin{aligned}
 r_1 : \quad & idle(x) \xrightarrow{offhook(x)} dialtone(x) \\
 r_2 : \quad & dialtone(x), idle(y) \xrightarrow{dial(x,y)} calling(x, y)
 \end{aligned}$$

Next, consider the following two states:

$$\begin{aligned}
 s &= idle(A), idle(B), dialtone(C), idle(D) \\
 s' &= idle(A), idle(B), calling(C, D)
 \end{aligned}$$

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 $\mathcal{N}(\mathcal{S})$ is logically equivalent to \mathcal{S} with respect to reachability analysis.

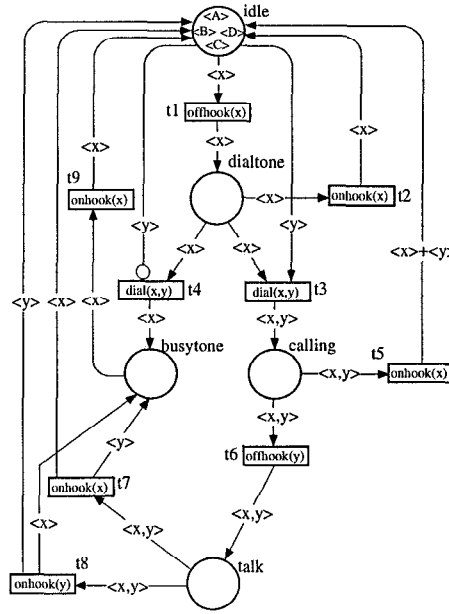


Figure 3: Service specification net for POTS

Lemma 1 For a given service specification \mathcal{S} and a service specification net $\mathcal{N}(\mathcal{S})$ for \mathcal{S} , there exists one-to-one correspondence between a set of reachable markings of $\mathcal{N}(\mathcal{S})$ and a set of reachable states of \mathcal{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_{p_2} of states which satisfy Condition P2(Phase 1) and then check reachability of each state in S_{p_2} (Phase 2 and Phase 3).

The set S_{p_2} can be easily calculated using the rules of service specification. Intuitively, a state in S_{p_2} 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_{p_2} . On performing this checking, we utilize extensively *P-invariant* of the service specification net.

6.2 P-Invariant

Definition 8 [6] Let $\mathcal{N}(\mathcal{S})$ be a service specification net with u places and v transitions, and let $p \in P$, $t \in T$ with $(p, t) \notin H$. Then, $W(p, t)$ (or $W(t, p)$) is a linear function $[C(t) \rightarrow C(p)]$ such that $\forall c = \langle a_1, \dots, a_l \rangle \in C(t)$, $W(p, t)(c) = L_a(p, t)\theta(c)$ (or,

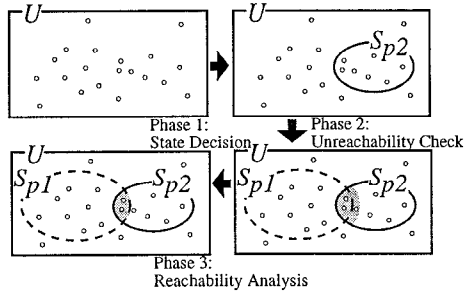


Figure 4: Outline of our method

$W(t, p)(c) = L_a(t, p)\theta(c)$, respectively). In our discussion, we consider only four kinds of linear functions defined as follows: (1) identity function id , (2) zero function o , (3) projection functions p_1 and p_2 such that $p_1(\alpha, \beta) = \langle \alpha \rangle$, $p_2(\alpha, \beta) = \langle \beta \rangle$.

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

$$A[p, t] = W(t, p) - W(p, t)$$

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

$$\begin{array}{c}
 \begin{matrix} idl \\ dil \\ clg \\ bst \\ tlk \end{matrix} \\
 \left[\begin{array}{ccccccccc}
 t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \\
 -id & id & -p_2 & o & p_1 + p_2 & o & p_1 & p_2 & id \\
 id & -id & -p_1 & -p_1 & o & o & o & o & o \\
 o & o & id & o & -id & -id & o & o & o \\
 o & o & o & p_1 & o & o & p_2 & p_1 & -id \\
 o & o & o & o & o & id & -id & -id & o
 \end{array} \right]
 \end{array}$$

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

$$Y = \begin{pmatrix} idle & dialtone & calling & busytone & talk \\ id & id & p_1 + p_2 & id & p_1 + p_2 \end{pmatrix}$$

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 $Y * M^t = Y * M_0^t$.

Remark 3 The equation $Y * M^t = Y * M_0^t$ is only a necessary condition for any reachable marking M . Hence, even if $Y * M^t = Y * M_0^t$ holds, we cannot conclude, in general, that M is reachable.

6.3 Analysis Algorithm Ω

This subsection shows the analysis method of non-determinism. The input of the algorithm is a service specification $\mathcal{S} = \langle R, s_0 \rangle$. Figure 5 shows the proposed algorithm Ω . In the following, we briefly explain Ω .

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_2} is a set of markings each of which corresponds to a state in S_{p_2} (defined in Section 4).

Next, Phase 1 determines a set of states S_{p_2} satisfying Condition P2. At first, we construct a condition C to which two rules r_i and r_j 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 $M \in MS_{p_2}$, if M is not reachable from M_0 by solving the equation $Y * M^t = Y * M_0^t$. According to Theorem 1, if the equation is not solvable (i.e., there is no assignment of U to wild-cards which satisfy $Y * M^t = Y * M_0^t$), then M is not reachable. Hence, we delete M from MS_{p_2} . If the equation is solvable, then we do not derive any decision on the reachability of M , thus we leave M in MS_{p_2} .

Finally in Phase 3 we must apply conventional reachability analysis method to the resultant MS_{p_2} (i.e., the size of MS_{p_2} 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_{p_2} in Phase 2.

Example 9 We apply the algorithm Ω 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:

$$Y = \begin{pmatrix} idle & dialtone & calling & busytone & talk \\ id & id & p_1 + p_2 & id & p_1 + p_2 \end{pmatrix}$$

Phase 1(Decision of States in S_{p_2}):

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

$$\begin{aligned}
 r_1 &: idle(x) \xrightarrow{offhook(x)} dialtone(x) \\
 r_6 &: calling(x, y) \xrightarrow{offhook(y)} talk(x, y)
 \end{aligned}$$

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

$$\begin{aligned}
 r_1\theta_1 &: idle(A) \xrightarrow{offhook(A)} dialtone(A) \\
 r_6\theta_6 &: calling(x, A) \xrightarrow{offhook(A)} talk(x, A)
 \end{aligned}$$

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

$$C = idle(A), calling(x, A)$$

Analysis algorithm Ω :

Phase 0(Preliminary): Construct the service specification net $\mathcal{N}(\mathcal{S})$ for a given service specification $\mathcal{S} = \langle R, s_0 \rangle$. Then calculate P-invariant Y of $\mathcal{N}(\mathcal{S})$. Define a set MS_{p_2} to be a set of markings, each of which corresponds to a state in S_{p_2} , and make the initial value of MS_{p_2} to be empty.

Phase 1(Decision of States in S_{p_2}):

Step1: Select two rules r_i and r_j from R whose event symbols are identical.

Step2: Apply a pair of substitutions θ_i and θ_j such that $e[r_i\theta_i] = e[r_j\theta_j]$ to r_i and r_j , respectively.

Step3: 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 mutual exclusive with each other, and go to Step 5. Otherwise, go to Step 4.

Step4: At first, put tokens to places based on predicates in C . Then, to each place $p \in P$, put the *wild-card* of tokens $\Sigma\langle x_{p_1}, \dots, x_{p_k} \rangle$, where x_{p_i} is a variable and k is arity of p . Construct a marking M_c which corresponds to the resulting net, and put M_c into MS_{p_2} .

Step5: If some pairs of rules to be checked still remain, then go to Step1.

Phase 2(Check of Unreachability using P-invariant): Check if for each marking $M \in MS_{p_2}$ is reachable by solving the equation $Y * M^t = Y * M_0^t$. 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 Ω

Step4: At first, we obtain a marking M according to C as follows:

$$M = \begin{array}{ccccc} \text{idle} & \text{dialtone} & \text{calling} & \text{busytone} & \text{talk} \\ \{\langle A \rangle\}, & \phi, & \{\langle x, A \rangle\}, & \phi, & \phi \end{array}$$

Next, we put five wild-cards to places, and finally get the following marking M_c :

$$M_c = \begin{array}{ccc} \text{idlc} & \text{dialtone} & \text{calling} \\ \{\langle A \rangle, \Sigma\langle x_1 \rangle\}, & \{\Sigma\langle x_2 \rangle\}, & \{\langle x, A \rangle, \Sigma\langle x_3, x_4 \rangle\}, \\ \text{busytone} & \text{talk} & \\ \{\Sigma\langle x_5 \rangle\}, & \{\Sigma\langle x_6, x_7 \rangle\} & \end{array}$$

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 get¹

$$\begin{aligned} Y * M_0^t &= id(\{\langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle\}) + id(\phi) + \\ &\quad (p_1 + p_2)(\phi) + id(\phi) + (p_1 + p_2)(\phi) \\ &= \{\langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle\}, \\ Y * M^t &= id(\{\langle A \rangle, \Sigma\langle x_1 \rangle\}) + id(\{\Sigma\langle x_2 \rangle\}) + (p_1 + p_2) \\ &\quad (\{\langle x, A \rangle, \Sigma\langle x_3, x_4 \rangle\}) + \\ &\quad id(\{\Sigma\langle x_5 \rangle\}) + (p_1 + p_2)(\{\Sigma\langle x_6, x_7 \rangle\}) \end{aligned}$$

¹In [6], the definition of linear functions is extended for the multi-set. For any linear function f , we define $f(\{\langle \alpha_1, \beta_1 \rangle, \dots, \langle \alpha_k, \beta_k \rangle\}) = \{f\langle \alpha_1, \beta_1 \rangle, \dots, f\langle \alpha_k, \beta_k \rangle\}$.

$$= \{2\langle A \rangle, \langle x \rangle, \Sigma\langle x_1 \rangle, \Sigma\langle x_2 \rangle, \Sigma\langle x_3 \rangle, \Sigma\langle x_4 \rangle, \Sigma\langle x_5 \rangle, \Sigma\langle x_6 \rangle, \Sigma\langle x_7 \rangle\}$$

For this, no matter how we nicely choose assignment of constant such as $\langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle$ to $\langle x \rangle$ and $\Sigma\langle x_i \rangle$'s, the equation $Y * M^t = Y * M_0^t$ 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 Ω , 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 Ω (Phase 0,1,2 and 3) identifies $S_{p_1} \cap S_{p_2}$.

Next, let us briefly estimate the cost of the algorithm Ω . In the following, m and n denote the number of rules and the number of users, respectively.

Cost of Phase0(C_0): The cost of transformation of a given service specification into the net model is obviously 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, $C_0 = m + c^m \approx c^m$.

Cost of Phase1(C_1): 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 Step1 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, C_1 is approximately estimated as $m * (m - 1) / 2 \approx m^2$.

Cost of Phase2(C_2): To check if a marking M is reachable or not, we have to evaluate $Y * M^t$ and $Y * M_0^t$. 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 of m . Finally, this checking must be repeated for all M 's in MS_{p_2} , thus $C_2 = m^c * C_1 = m^c * m^2 \approx m^c$.

Cost of Phase 3(C_3): C_3 deeply depends on the size of resultant MS_{p_2} , and it is generally exponential order of n . Since MS_{p_2} seems to depend on the service specification, C_3 cannot be evaluated reasonably without experimental application to many practical services(Surely if $MS_{p_2} = \phi$ then $C_3 = 0$ as shown in Example 9).

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 Ω (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 c^m 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 unreachability(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 Ω to practical services[11, 12], and are developing a computer aided tool for the experiments.

Acknowledgement

This reasearch 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.), Mr. Y.Wakahara(KDD), and research staffs in KDD R&D Laboratories for their helpful discussion.

References

- [1] Alla, H., Ladet, P., Martinez and J., Silva-Suarez, M., "Modelling and validation of complex systems by coloured Petri-nets: Application to a flexible manufacturing system," *Lecture Notes in Computer Science*, Vol 188, Springer-Verlag, pp.215-233, 1985.
- [2] Cameron, E.J. and Velthuijsen, H., "Feature interactions in telecommunications systems," *IEEE Communication Magazine*, Vol.31, No.8, pp.18-23, 1993.
- [3] Cameron, E.J., Griffeth, N.D., Lin, Y-J., Nilson, M.E., Schure W.K. and Velthuijsen, H., "A feature interaction benchmark for IN and Beyond," *Proc. of Second Workshop on Feature Interactions in Telecommunications Systems*, pp.1-23, 1994.
- [4] Gammelgaard, A. and Kristensen E.J., "Interaction detection, a logical approach," *Proc. of Second Workshop on Feature Interactions in Telecommunications Systems*, pp.178-196, 1994.
- [5] Harada, Y., Hirakawa, Y., Takenaka, T. and Terashima, N., "A conflict detection support method for telecommunication service descriptions," *IEICE Trans. Commun.*, Vol. E75-B, No.10, Oct, 1992.
- [6] Jensen, K., "Coloured Petri Nets," *EATCS Monographs on Theoretical Computer Science*, Vol1-2, Springer Verlag, 1992.
- [7] Hirakawa, Y. and Takenaka, T., "Telecommunication service description using state transition rules," *Proc. of IEEE Int'l Workshop on Software Specification and Design*, pp.140-147, Oct, 1991.
- [8] Murata, T. and Zhang, D., "A predicate-transition net model for parallel interpretation of logic programs," *IEEE Trans. on Software Engineering*, Vol.14, No.4, pp.481-497, Apr. 1988.
- [9] Ohta, T. and Harada Y., "Classification, detection and resolution of service interactions in telecommunication services," *Proc. of Second Workshop on Feature Interactions in Telecommunications Systems*, pp.60-72, 1994.
- [10] Wakahara, Y., Fujioka, M., Kikuta, H., Yagi, H. and Sakai, S., "A method for detecting service interactions," *IEEE Communication Magazine*, Vol.31, No.8, pp.32-37, 1993.
- [11] ITU-T Recommendations Q.1200 Series., "Intelligent Network Capability Set 1 (CS1)", Sept. 1990.
- [12] Bellcore, "LSSGR Features Common to Residence and Buisness Customers I, II, III," Issue 2, July 1987.