Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 328 (2004) 245–269

Theoretical Computer Science

www.elsevier.com/locate/tcs

Structure and behavior preservation by Petri-net-based refinements in system design

Hejiao Huang[a,b,*], To-yat Cheung[b], Wai Ming Mak[b]

[a] *Department of Mathematics, Shaan Xi Normal University, Xi'an, China*

[b] *Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, China*

Received 9 December 2003; received in revised form 15 June 2004; accepted 7 July 2004

Communicated by D.-Z. Du

Abstract

A refinement is a transformation for replacing a simple entity of a system with its functional and operational details. In general, the refined system may become incorrect even if the original system is correct because some of its original properties may have been lost or some unneeded properties may have been created. For systems specified in pure ordinary Petri nets, this paper proposes the conditions imposed on several types of refinement under which the following 19 properties will be preserved: state machine, marked graph, free choice net, asymmetric choice net, conservativeness, structural boundedness, consistence, repetitiveness, rank, cluster, rank-cluster-property, coverability by minimal state-machines, siphon, trap, cyclomatic complexity, longest path, boundedness, liveness and reversibility. Such results have significance in three aspects: (1) It releases the designer's burden for having to provide different methods for individual properties. (2) In the literature, refinements have been shown preserving several equivalence relations and behavioral properties. Our results show that they also preserve many structural properties. (3) It greatly enlarges the scope of applicability of refinements because they can now be applied on systems that satisfy more properties than just liveness and boundedness.

Keywords: Design; Petri net; Preservation; Property; Refinement; Specification; Verification

* Corresponding author. Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong, China.

E-mail addresses: hejiao@cs.cityu.edu.hk (H. Huang), cscheung@cityu.edu.hk (T.-y. Cheung).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.

doi:10.1016/j.tcs.2004.07.016

1. Introduction

In Petri-net-based modular system design, transformations, such as refinements, compositions, reductions, etc., are often applied on the nets in order to develop a correct design specification. In this process, one of the difficult tasks is to verify that the transformed nets are correct, i.e., possessing certain desirable properties. In the literature, there exist three popular approaches for such purposes. The *first* approach may be called direct proof where by the transformed net is directly shown (i.e., based on definitions or by constructive algorithms) whether possessing the desirable properties or not. The challenge of this approach is that distinct verification methods may have to be sought for different nets and different properties. The *second* approach is called property-preservation. In this approach, the original net is assumed to be satisfying some specific properties and the transformation is required to preserve these properties in the transformed net. The advantage of this approach is that the transformed net is automatically correct without the need of further verification. However, the challenge is to find the appropriate transformation that preserves the specified properties. The *third* approach is to use characterizations. A characterization is a relationship that relates several properties. Sometimes, for a specific property, it may be too difficult to find a direct proof or no transformations that preserve this property are available. Then, based on some known characterizations, a designer may verify the other properties appearing in the characterization instead of verifying the specified one. For example, based on Characterization AC-5 of Section 2, in order to prove the liveness of a general Petri net, it is sufficient to prove that it satisfies several structural properties.

This paper investigates a special type of transformations called refinement and its property-preserving approach for verification. It shows that several elementary refinements can preserve 19 properties. These 19 properties can accommodate many aspects of systems. As explained below, such results constitute a significant contribution to the application and theory of Petri-net-based transformations for system design.

(a) *Our results extend the preservation of behavioral properties to structural properties:* In Petri-net-based system design, properties are the backbone for many aspects of investigation. In the olden days, knowledge in this field was limited to a few behavioral properties, such as liveness and boundedness. At present, these few properties are inadequate for designing complex real-life systems wherein a great variety of system-dependent and domain-dependent properties are involved. As this field is becoming more mature, the number of properties under study has been greatly increased. Many new structural properties and behavioral properties [11] have been reported in the literature.

In the survey paper, Brauser et al. [1] classified refinement techniques into two categories according to what kind of properties they preserve. The first category preserves behavioral properties (which include boundedness and liveness), whereas the second category preserves semantic equivalence (e.g., failure equivalence). Most of the refinement techniques reported in the literature (including [16,17]) belong to the first category. However, most of them considered the preservation of liveness and boundedness only. This paper follows the recent research trend of this field. It shows that, besides the three common behavioral properties boundedness, liveness and reversibility, the various refinements can also preserve 16 structural properties, namely, state machine, marked graph, free-choice-ness, asymmetric-choice-ness, conservativeness, structural boundedness, consistence, repetitive-

ness, rank, cluster, rank-cluster-property, coverability by minimal state-machines, siphon, trap, cyclomatic complexity and longest path.

(b) *Our results enhance the characterization-based approach for system verfication:* Sometimes, in system verification, it is too difficult to verify directly a specific property or to show that a specific property is preserved. However, if some algebraic characterizations relating this property with some others are available, it may be simpler to do this for the other properties than for the specific property. Hence, if more properties are involved, it may be easier to find the appropriate characterizations or property-preserving transformations.

(c) *Our results enhance the property-preserving approach for system verification:* In system design, a transformation may serve two purposes. One is to *permanently modify* the system according to the objective in the design process. The other is to *temporarily modify* the system so that it is easier to verify certain properties of the original system by verifying the modified system. The transformation is temporary because, after the verification, it will be abandoned and the design process resumes from the original net.

For the first purpose, since a transformation is permanent, a single transformation should be able and enough to 'carry' *all* the properties from the original net to the transformed net. For example, suppose we want to create a live system that can be covered by state machines (SM-coverable). If we start with a draft live net that is SM-coverable, we should apply only those refinements that can preserve both liveness and SM-coverability. Hence, it is necessary to show that the refinement, besides liveness, also preserves SM-coverability. In general, the more properties a refinement is shown to preserve, the more advanced systems it can be applied to design.

For the second purpose, if a single transformation cannot preserve all the specified properties, a designer has to apply several different transformations which can preserve these properties separately. Though being able to achieve the goal of verification if such transformations are available, obviously it will not be so efficient as if applying a single transformation that can preserve all these properties.

In the early stage of research in property-preserving transformations, the scope of studies was largely quite narrow. For instance, existing results concerning refinements and reductions are mostly about liveness and boundedness and seldom about other properties. Also, most transformations are quite specific, such as merging a few places or transitions [9–11], reducing [7,13] and refining [16,17] individual places or transitions or very specific subnets. To circumvent these shortcomings, current research is expanding in two directions, one aiming at properties other than liveness and boundedness and another at more general transformations. For instance, Cheung, Zeng and Lu provided conditions for the preservation of place-invariants under five very general transformations on ordinary Petri nets [2] and colored Petri nets [3]. Mak [11] investigated the preservation of 20 properties under many transformations defined in terms of various operations (such as choice, sequential, interleave, parallel, disable, eliminations, etc.) on software processes.

Note that this paper does not propose totally new refinement techniques but focuses on the expansion of properties they can preserve. For example, as will be described below, there are some minor differences between the refinement techniques of Valette [17] and Susuki et al. [16] (referred to as VS below) and ours. In fact, the differences lie just in the ways of modeling the refinement net and stating the assumptions. However, these minor differences have led to different ways for proving those common results—the preservation

of liveness and boundedness. The major difference between these two papers and ours is that ours shows that, in addition to the few behavioral properties, refinements can also preserve many structural properties.

(a) Our refinement net B_2 has a unique entry place with multiple exit transitions and a unique exit place with multiple entry transitions. In VS's models, the refinement net starts with a unique entry transition and ends with a unique exit transition. However, from the viewpoint of system structures, if we ignore the system properties our model and VS's models can be easily converted to each other.

(b) In order for a refinement to work properly, B_2 should be non-re-enterable. That is, once started, B_2 cannot be initiated again until the current cycle has terminated. In VSs' models, this is ensured by requiring B_2 to be well-formed or k-well-behaved and the refined transition t_r not to be 2-enabled or $(k+1)$-enabled in the refined net. In our model, this is ensured by requiring B_2 to initiate and terminate properly and t_r not to be 2-enabled.

The rest of the paper is organized as follows. Section 2 presents some basics about Petri nets including Petri net processes and algebraic characterizations. Transition and place refinements and their preservation of 19 properties are presented in Sections 3 and 4, respectively. In Section 5, an example is given. Some concluding remarks are given in Section 6. In order not to interrupt the flow of description in the main context of the paper and to provide convenient references to the readers, theorem proofs are put in the appendix.

2. Basics of Petri nets and Petri net processes

This section presents some basics of Petri nets. In particular, it defines Petri net processes that are used to replace the specified transitions and some new properties specifically for them.

2.1. Petri net

A *net* is a 4-tuple $N = (P, T, F, W)$, where P is a finite set of *places*, T is a finite set of *transitions* such that $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation* and W is a *weight function* such that $W(x, y) \in \mathcal{N}^+$ (positive integers) if $(x, y) \in F$ and $W(x, y) = 0$ if $(x, y) \notin F$. A net is said to be *ordinary* if $W = 0$ or 1. In this case, W will be omitted.

The *marking* (or *state*) for a net is a function $M: P \to \mathcal{N}$ (non-negative integers). M may be represented by a multi-set expression or a $|P|$-vector $(M(p_1), \ldots, M(p_{|P|}))$, where $M(p)$ is the number of *tokens* in place $p \in P$. For example, if $M = (1, 0, 0, \ldots, 0)$, then the multi-set expression $M + 2p_2 + 3p_{|P|}$ represents the marking $(1, 2, 0, \ldots, 0, 3)$.

A *Petri net* (N, M_0) is a net with an initial marking M_0. A transition $t \in T$ is said to be *k-enabled* (or *k-firable*) for $k \in \mathcal{N}^+$ at a marking M iff $\forall p \in P: (M(p) \geqslant kW(p, t))$. k is omitted if being equal to 1. *Firing* (or *executing*) transition t results in changing marking M to marking M', where $M'(p) = M(p) - W(p, t) + W(t, p)\ \forall p \in P$.

Let (N, M_0) be a Petri net. N is said to be *pure* or *self-loop-free* iff $\forall x, y \in P \cup T$: $((x, y) \in F \Rightarrow (y, x) \notin F)$. $M[N, t\rangle$ means that transition t is enabled at M. For a sequence $\sigma = t_1 \ldots t_k \in T^*$, $M[N, \sigma\rangle$ means that there exist markings $M_i, i = 1, \ldots, k$ such that $M_0 = M$ and $M_{i-1}[N, t_i\rangle M_i$ and $M_{k-1}[N, t_k\rangle$. $L(B, M_0)$ denotes the language of (B, M_0), i.e., $\mathrm{L}(B, M_0) = \{\sigma | M[N, \sigma\rangle\}$. $M[N, \sigma\rangle M'$ means that M' is *reachable* from M by firing sequence σ. If σ is not explicitly specified, the notation $M[N, *\rangle M'$ is used. $R[N, M]$ denotes the *reachability set* of N from marking M, i.e., the smallest set of markings such that (i) $M \in R[N, M]$ and (ii) if $M' \in R[N, M]$ and $M'[N, t\rangle M''$ for some $t \in T$, then $M'' \in R[N, M]$. N may be omitted if understood.

$V = \mathrm{POST}–\mathrm{PRE}$ is the *incidence matrix* of N, where PRE is a $|P| \times |T|$ matrix whose element $w(p, t)$ is the weight of the arc from place p to transition t. POST is a $|P|^\bullet \times |T|$ matrix whose element $w(t, p)$ is the weight of the arc from transition t to place p. The rank of V may be written as $\mathrm{Rank}(V)$ or $\mathrm{Rank}(N)$. For matrix V with rows in P and columns in T, if $P_1 \subseteq P$ and $T_1 \subseteq T$, then $V[P_1, T_1]$ denotes the sub-matrix of V with rows in P_1 and columns in T_1. $V[p, T]$, $V[P, t])$ and $V[p, t]$ are particular cases for row p and column t. Also, if it is clear from the context, we do not distinguish the symbols between column vectors and row vectors. For a Petri net (N, M_0), $M = M_0 + V\mu$, is called the *state equation*, where μ is called the *firing count vector* of a firing sequence σ if $\mu[t] = \#(\sigma, t)$ is the number of times transition t occurs in σ. For $x \in P \cup T$, ${}^\bullet x = \{y | (y, x) \in F\}$ and $x^\bullet = \{y | (x, y) \in F\}$ are called the *pre-set* (*input set*) and *post-set* (*output set*) of x, respectively. For a set $X \subseteq P \cup T$, ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. $\#(\sigma, p^\bullet) = \sum_{t \in p^\bullet} \#(\sigma, t)$, $\#(\sigma, {}^\bullet p) = \sum_{t \in {}^\bullet p} \#(\sigma, t)$.

The definitions of some basic structural and behavioral properties of a Petri net (N, M_0) referenced in this paper are listed below.

- A place p is said to be *bounded* (resp., *k-bounded*) iff $\exists k > 0$: $(\forall M \in R[N, M_0\rangle, M(p) \leqslant k]$. (N, M_0) is said to be *bounded* (resp., *k-bounded*) iff every place of N is bounded (resp., k-bounded). A place is said to be *safe* iff it is 1-bounded. (N, M_0) is said to be *safe* iff all its places are 1-bounded. N is said to be *structurally bounded* iff it is bounded for every initial marking M_0.
- For $x \in P \cup T$, the *cluster* [6] of x, denoted as $[x]$, is the smallest subset of $P \cup T$ satisfying three conditions: (1) $x \in [x]$; (2) if $p \in P \cap [x]$ then $p^\bullet \subseteq [x]$; and (3) if $t \in T \cap [x]$ then ${}^\bullet t \subseteq [x]$. The *set of clusters* of N is denoted as $\mathrm{C}(N) = \{[x] | x \in P \cup T\}$.
- N is *connected* [6] iff it is not composed of two disjoint and non-empty subnets. N is *strongly connected* if and only if, for every pair of nodes x and y, there exists a directed path from x to y.
- N is *conservative* [15] iff there exists a $|P|$-vector $\alpha \geqslant 1$ such that $\alpha V = 0$, *consistent* [15] iff there exists a $|T|$-vector $\beta \geqslant 1$ such that $V\beta = 0$ and *repetitive* [12,15] iff there exists a $|T|$-vector $\beta \geqslant 1$ such that $V\beta \geqslant 0$.
- A transition t is said to be *live* iff $\forall M \in R[N, M_0\rangle$: $(\exists M'$: $(M[*\rangle M'$ and $M'[t\rangle)$, or equivalently, $\forall M \in R[N, M_0\rangle$: $(\exists \sigma \in T^*$: $(M[\sigma t\rangle)$. (N, M_0) is said to be *live* iff every transition of N is live. N is said to be *structurally live* if there exists a marking M_0 such that (N, M_0) is live [5,8].
- N is said to satisfy the *rank-and-cluster property* (*RC-property*) [15] if $\mathrm{Rank}(N) = |C(N)| - 1$.

- (N, M_0) is said to be *reversible* [5] if $\forall M \in R[N, M_0\rangle$: $(M_0 \in R[N, M\rangle$.
- A set of places D is called a *siphon* (or *deadlock*) if ${}^\bullet D \subseteq D^\bullet$ and a *trap* if $D^\bullet \subseteq {}^\bullet D$ [5].
- A subnet $N_i = (P_i, T_i, F_i)$ of N is an *SM-component* of N if N_i is a strongly connected SM and $T_i = {}^\bullet P_i \cup P_i^\bullet$. N is said to be *SM-coverable* [6] iff there exists a set of *SM-components* $\{N_1, \ldots, N_k\}$ such that $P = \cup_i P_i$, $T = \cup_i T_i$ and $F = \cup_i F_i$, where i runs from 1 to k. $\{N_1, \ldots, N_k\}$ is called an *SM-cover* of N and is said to be *minimal* iff none of its proper subsets is also an SM-cover of N.
- N is called a *state machine* (*SM*) [6,12] iff $\forall t \in T$: $(|{}^\bullet t| = |t^\bullet| = 1)$, a *marked graph* (*MG*) iff $\forall p \in P$: $(|{}^\bullet p| = |p^\bullet| = 1)$, a *free choice* (*FC*) net iff $\forall t_1, t_2 \in T$: $({}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow {}^\bullet t_1 = {}^\bullet t_2)$, and an *asymmetric choice* (*AC*) net iff $\forall t_1, t_2 \in T$: $({}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset \Rightarrow {}^\bullet t_1 \subseteq {}^\bullet t_2$ or ${}^\bullet t_2 \subseteq {}^\bullet t_1)$.
- N is said to be *well-formed* if there exists an M_0 such that (N, M_0) is live and bounded.

Some algebraic characterizations of Petri nets are summarized as follows:

AC-1 (*Necessary condition for a reachable marking*). Let α be a place invariant of (N, M_0). Then, $\forall M \in R[N, M_0\rangle$, $\alpha M = \alpha M_0$.

AC-2 (*Sufficient condition for repetitiveness* [12]). A consistent Petri net is repetitive.

AC-3 (*Sufficient condition for strong connectedness*, [8]). A conservative, consistent and connected net is strongly connected.

AC-4 A net is structurally bounded [12] iff $\exists$|P|-vector $\alpha \geqslant 1$: $(\alpha V \leqslant 0)$.

AC-5 (*Rank Theorem for liveness*, [6]). Let N be an ordinary Petri net which is connected, conservative, consistent and satisfies the RC-property. (N, M_0) is live if $\alpha M_0 > 0$ for every semi-positive place invariant α of (N, M_0).

AC-6 (*Rank Theorem*, [6]). Let N be an ordinary Petri net. If N is connected, conservative, consistent and satisfies the RC-property, then N is well-formed. If N is well-formed, then $\mathrm{Rank}(N) < |\{{}^\bullet t \mid t \in T\}|$.

AC-7 (*Sufficient condition for FC net*). If $\forall p \in P$: ${}^\bullet(p^\bullet) = \{p\}$), then N is a FC net.

AC-8 (*Rank Theorem for FC net*, [6]). Let N be a FC Petri net. N is well-formed iff N is connected, conservative, consistent and satisfies the RC-property.

AC-9 (*Sufficient condition for SM-coverability*, [6]). Let N be an ordinary Petri net. If N is connected, conservative, consistent and satisfies the RC-property, then N is SM-coverable.

Throughout this paper, we consider only pure, ordinary Petri nets.

2.2. Petri net process

In a transition refinement, a transition is replaced with a Petri net process, a type of nets with a special structure, a control marking and some initiation/termination rules.

Definition 2.1 (*Process*). A *net process* (or just *process*) is a 3-tuple $B = (N, p_e, p_x)$, where

(a) $N = (P, T, F)$ is an ordinary, pure and connected net,
(b) p_e (the *entry place*) is the only place $p \in P$ such that ${}^\bullet p = \emptyset$ and
(c) p_x (the *exit place*) is the only place $p \in P$ such that $p^\bullet = \emptyset$.

A *Petri net process* (B, M_c) is a process B with a *control marking* M_c satisfying the constraints: $M_c(p_e) = M_c(p_x) = 0$. (B, M_c) satisfies two assumptions:

(a) *Proper initiation*: B can be initiated only at the *entry marking* $M_e = p_e + M_c$.

(b) *Proper termination*: B terminates on and only on reaching any marking M for which $M(p_x) \geqslant 1$. Furthermore, $\forall M \in [B, M_e\rangle$, where $M(p_x) \geqslant 1$, M can only be the *exit marking* $M_x = p_x + M_c$.

Proper initiation ensures that a process can be initiated *only* by depositing a token into p_e. Proper termination, however, does not imply that a process *must* terminate (which is another design issue). It only means that, whenever p_x has a token, then B must terminate; and that, if B terminates, its marking must be M_x. These assumptions arise out of the activation and memorylessness requirements in the management of software processes. They serve similar purposes as the assumptions of well-formedness [17] and 1-well-behavedness [16]. Extending our model to k-well-behavedness is quite straightforward.

Definition 2.2 (*Associated process, re-initiation path, 'Almost' properties*). The *associa ted process* B_a of process B is formed from B by adding an *associated transition* t_a and two arcs (p_x, t_a) and (t_a, p_e) to N. The sequence $p_x t_a p_e$ is called the *reinitiation path.* Process B is said to *almost satisfy Property X* if B_a satisfies X.

B_a is introduced for describing some properties (e.g., SM-coverability defined in the appendix) of B that are based on strong connectedness, since B is not strongly connected but B_a is. For example, B is *almost a marked graph* (*MG*) if B_a is an MG. B is *almost SM-coverable* if B_a is SM-coverable.

Definition 2.3 (*Elementary entry-to-exit path, sequence (e-path, e-sequence)*). $\gamma = t_1 \ldots t_j \ldots t_n \in T^*$ is called an *elementary entry-to-exit path* or sequence (*e-path* or *e-sequence*) if $t_1 \in p_e^\bullet, t_n \in {}^\bullet p_x, t_j \notin p_e^\bullet \cup {}^\bullet p_x$ for $j = 2, \ldots, n-1, t_i^\bullet \cap {}^\bullet t_{j+1} \neq \phi$ for $j = 1, \ldots, n-1$, and $t_i \neq t_j$ if $i \neq j$. The length of the *longest e-path* of a process B is denoted as $\mathrm{LP}(B) = \max\{k | k = |\gamma|, \gamma$ is an e-path of B$\}$.

Definition 2.4 (*Cyclomatic complexity, Cheung and Zu [3]*). The *cyclomatic complexity* of a Petri net process B without the re-initiation path is defined as $Z(B) = |F| - |P \cup T| + 2$, where $|F|$ is the total number of arcs and $|P \cup T|$ is the total number of places and transitions of B. If B has the initiation path (i.e., B_a), $Z(B_a) = |F| - |P \cup T| + 1$.

Cyclomatic complexity [14] is a well-known quantitative measure of software programs that can be represented as a program graph. In the basis path testing methods, it is an upper bound on the number of independent e-paths that cover every operation of the program at least once. Cheung et al [4,11] extended cyclomatic complexity from program graphs to Petri net processes and proved its preservation under many transformations. The number LP(B) may be used to estimate the maximum number of distinct operations needed for executing the process once. However, note that an e-sequence is not necessarily firable.

3. Property-preserving refinements of transitions

A transition refinement expands a transition to a Petri net process. This section first formally describes the refinement technique and then derives some relationships among its

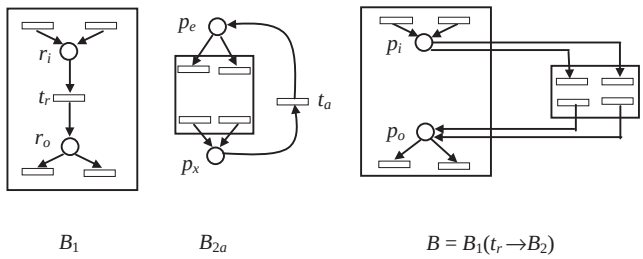

Fig. 1. Transformation TR (Transition Refinement).

various ingredients. Then, one of the two major results of this paper, namely, preservation of 19 properties under a transition refinement, is presented (Theorems 3.3 and 3.4).

Definition 3.1 (*Refined net, refined transition (B_1 in Fig. 1)*). A *refined net* is a 4-tuple $B = (N, r_\mathrm{i}, r_\mathrm{o}, t_\mathrm{r})$, where

- $N = (P, T, F)$ is an ordinary net,
- $r_\mathrm{i} \in P$ is the *refinement inlet place*, $r_\mathrm{o} \in P$ is the *refinement outlet place* and $t_\mathrm{r} \in T$ is the *refined transition*, where $r_\mathrm{i}^\bullet = \{t_\mathrm{r}\} = {}^\bullet r_\mathrm{o}$ and $|{}^\bullet t_\mathrm{r}| = |t_\mathrm{r}^\bullet| = 1$,
- t_r is not 2-enabled.

A *marked refined net* (B, M_r) is a refined net with an initial marking M_r.

Transformation TR (*Transition Refinement $B_1(t_\mathrm{r} \rightarrow B_2)$*)(Fig. 1). Let (B_1, M_r) be a marked refined net and (B_2, M_c) be a proper terminated Petri net process. Replacing transition t_r of (B_1, M_r) with (B_2, M_c) results in a marked net (B, M_i), where

- $B = B_1(t_\mathrm{r} \rightarrow B_2) = (N, p_\mathrm{i}, p_\mathrm{o})$; $N = (P, T, F)$.
- $P = P_1 \cup P_2 \cup \{p_\mathrm{i}, p_\mathrm{o}\} - \{r_\mathrm{i}, r_\mathrm{o}, p_\mathrm{e}, p_\mathrm{x}\}$; $T = T_1 \cup T_2 - \{t_\mathrm{r}\}$; $F = F_1 \cup F_2 \cup (\{p_\mathrm{i}, x) | x \in p_\mathrm{e}^\bullet\} \cup \{(x, p_\mathrm{o}) | x \in {}^\bullet p_\mathrm{x}\} \cup \{(x, p_\mathrm{i}) | x \in {}^\bullet r_\mathrm{i}\} \cup \{(p_\mathrm{o}, x) | x \in r_\mathrm{o}^\bullet\}) - (\{(r_\mathrm{i}, t_\mathrm{r}), (t_\mathrm{r}, r_\mathrm{o})\} \cup \{(x, r_\mathrm{i}) | x \in {}^\bullet r_\mathrm{i}\} \cup \{(r_\mathrm{o}, x) | x \in r_\mathrm{o}^\bullet\} \cup \{(p_\mathrm{e}, x) | x \in p_\mathrm{e}^\bullet\} \cup \{(x, p_\mathrm{x}) | x \in {}^\bullet p_\mathrm{x}\})$.
- p_i, is the *inlet* place and p_o is the *outlet* place
- The initial marking M_i of B is derived from M_r of B_1 and M_c of B_2 as follows: $M_\mathrm{i}(p_\mathrm{i}) = M_\mathrm{r}(r_\mathrm{i})$; $M_\mathrm{i}(p_\mathrm{o}) = M_\mathrm{r}(r_\mathrm{o})$; $M_\mathrm{i}(p) = M_\mathrm{r}(p)$ for $p \in P_1 - \{r_\mathrm{i}, r_\mathrm{o}\}$ and $M_\mathrm{i}(p) = M_\mathrm{c}(p)$ for $p \in P_2 - \{p_\mathrm{e}, p_\mathrm{x}\}$.

For the rest of this paper, B_1 denotes the refined net, B_2 the refinement net process and $B = B_1(t_\mathrm{r} \rightarrow B_2))$ the resulting net as described in Transformation TR.

Definition 3.2 (*Mappings arising from Transformation TR*). Let $\sigma \in L(B, M_\mathrm{i})$ and $M \in M_\mathrm{i}[B, \sigma\rangle$. The mappings of σ and M from B to B_1 and B_2 are defined below, where λ is a null sequence and γ is an e-sequence of B_2.

$f_1 \colon T^* \rightarrow T_1^*$ is defined as follows: $f_1(\lambda) = \lambda$. If $f_1(\sigma)$ has been defined for σ, then

$$\begin{aligned} f_1(\sigma t) &= f_1(\sigma) && \text{if } t \in T_2 - {}^\bullet p_\mathrm{o} \\ &= f_1(\sigma) t && \text{if } t \in T_1 - \{t_\mathrm{r}\} \\ &= f_1(\sigma) t_\mathrm{r} && \text{if } t \in {}^\bullet p_\mathrm{o}. \end{aligned}$$

M_1 is a restriction of M from P to P_1, where

$$\begin{aligned} M_1(r_\mathrm{i}) &= M(p_\mathrm{i}) + (\#(\sigma, p_\mathrm{i}^\bullet) - \#(\sigma, {}^\bullet p_\mathrm{o})), \\ M_1(r_\mathrm{o}) &= M(p_\mathrm{o}), \\ M_1(p) &= M(p) \quad \text{if } p \in P_1 - \{r_\mathrm{i}, r_\mathrm{o}\}. \end{aligned}$$

$f_2 : T^* \to T_2^*$ is defined as follows: $f_2(\lambda) = \lambda$. If $f_2(\sigma)$ has been defined for σ, then

$$\begin{aligned} f_2(\sigma t) &= f_2(\sigma) && \text{if } t \in T_1 - \{t_\mathrm{r}\} \\ &= f_2(\sigma)t && \text{if } t \in T_2 - {}^\bullet p_\mathrm{o} \\ &= \gamma && \text{if } t \in {}^\bullet p_\mathrm{o}. \end{aligned}$$

M_2 is a restriction of M from P to P_2, where

$$\begin{aligned} M_2(p_\mathrm{e}) &= M(p_\mathrm{i}), \\ M_2(p_\mathrm{x}) &= 0, \\ M_2(p) &= M(p) \quad \text{if } p \in P_2 - \{p_\mathrm{e}, p_\mathrm{x}\}. \end{aligned}$$

Lemma 3.1. *For any firable e-sequence γ of Petri net process (B, M_e), $M_\mathrm{e}[B, \gamma\rangle M_\mathrm{x}$.*

The following results describe the relationships among the transition sequences σ, $f_1(\sigma)$, $f_2(\sigma)$ and the markings M, M_1, M_2 of B, B_1 and B_2, respectively.

Lemma 3.2. *For any $\sigma \in L(B, M_\mathrm{i})$, $\#(\sigma, {}^\bullet p_\mathrm{o}) + 1 \geqslant \#(\sigma, p_\mathrm{i}^\bullet) \geqslant \#(\sigma, {}^\bullet p_\mathrm{o})$.*

Theorem 3.1. *Suppose a transition sequence σ and a marking M satisfy $M_\mathrm{i}[B, \sigma\rangle M$ in Transformation TR. Let f_1, f_2, M_1 and M_2 be defined as in Definition* 3.2. *Then, the following two propositions hold*: (1) $f_1(\sigma) \in \mathrm{L}(B_1, M_\mathrm{r})$ *and* $f_2(\sigma) \in \mathrm{L}(B_2, M_\mathrm{e})$. (2) $M_\mathrm{r}[B_1, f_1(\sigma)\rangle M_1$ *in* B_1 *and* $M_\mathrm{e}[B_2, f_2(\sigma)\rangle M_2$ *in* B_2.

Let us consider the implication of two cases of Theorem 3.1.

Case 1: $\#(\sigma, p_\mathrm{i}^\bullet) - \#(\sigma, {}^\bullet p_\mathrm{o}) > 0$. In this case, execution of B_2 has initiated but has not terminated yet. Furthermore, $M(p_\mathrm{i}) = 0$ but $M_1(r_\mathrm{i}) = 1$. This apparent inconsistence can be explained as follows. From the viewpoint of B, the token at p_i that initiates B_2 has been removed; whereas, from the viewpoint of B_1, this token till stays in r_i. Then, since t_r is not 2-enabled, B_2 cannot be re-entered.

Case 2: $\#(\sigma, p_\mathrm{i}^\bullet) - \#(\sigma, {}^\bullet p_\mathrm{o}) = 0$. In this case, execution of B_2 either has never occurred or has terminated. If the last transition t of σ is in ${}^\bullet p_\mathrm{o}$, then t is projected onto the single transition t_r within B_1 and the entire σ is the same as a firable e-sequence γ when observed within B_2. Furthermore, we have $M_1(r_\mathrm{o}) = M(p_\mathrm{o}) = 1$ but $M_2(p_\mathrm{x}) = 0$. This implies that any token in p_o is considered as belonging to B_1 rather than to B_2. That is, we have $M_\mathrm{e}[B_2, \gamma\rangle M_\mathrm{c}$ but $M_\mathrm{i}[B, \gamma\rangle M_\mathrm{o}$.

Lemma 3.3. (1) *Any $\mu \in L(B_2, M_\mathrm{e})$* is a *firable subsequence of some $\sigma \in (B, M_\mathrm{i})$ at a marking M where $M(p_\mathrm{e}) \geqslant 1$.* (2) *Suppose (B_2, M_e) has at least one firable e-sequence. Then, for any $\mu \in L(B_1, M_\mathrm{r})$, there exists $\sigma \in L(B, M_\mathrm{i})$ such that $\mu = f_1(\sigma)$.*

Theorem 3.2. *For the composite net obtained by Transformation TR, the following two propositions hold*: (1) B_2 *can be initiated only from* B_1. (2) *As far as token distribution in* B_1 *is concerned, executing one cycle of* B_2 *is equivalent to firing* t_r *once in* B_1.

In general, it is difficult to relate the ranks of the incidence matrices of the nets involved in Transformation TR. Theorem 3.3 provides such a relation for several important special cases.

Theorem 3.3 (*Relation of ranks of Petri nets in Transformation TR*). *Let* S, S', S_1 *and* S_2 *are S-invariants of* $B_1\backslash\{t_r\}$ *or* B_2. *Accordingly,* p_i *should be replaced by* r_i *or* p_e *and* p_o *by* r_o *or* p_x. *Then,* $\mathrm{Rank}(B_1(t_r \to B_2)) = \mathrm{Rank}(B_1) + \mathrm{Rank}(B_2) - h + d - 2$, *where*

$h = 0$ *if* t_r *appears in at least one T-invariant of* B_1
$\quad\;\; 1$ otherwise,

$d = 2$ *if* $\exists S_1, S_2$ *such that one of the following conditions holds*: (1) $p_i \in S_1$, $p_o \notin S_1$, $p_o \in S_2$. (2) $p_o \in S_2$, $p_i \notin S_2$, $p_i \in S_1$.
$\quad = 1$ *if one of the conditions holds*: (1) *One and only one of* $\{p_i, p_o\}$ *appears in some S-invariant*. (2) $\exists S_1$ *such that* $\{p_i, p_o\} \subseteq S_1$, *and* $\forall S' \neq S_1$, $S' \cap \{p_i, p_o\} = \emptyset$. (3) $B_1\backslash\{t_r\}$ *has an S-invariant* S_1 *of the form*: $a_1 p_i + b_1 p_o + c_1 = 0$ (*where* $a_1 \neq 0$ *and* $b_1 \neq 0$) *and* B_2 *has an S-invariant* S_2 *of the form*: $a_2 p_i + b_2 p_o + c_2 = 0$ (*where* $a_2 \neq 0$ *and* $b_2 \neq 0$) *such that* $b_2/a_2 = b_1/a_1$. *Furthermore,* $B_1\backslash\{t_r\}$ *and* B_2 *have no other S-invariants containing* p_i *or* p_o.
$\quad = 0$ *if neither* p_i *nor* p_o *appears in any S-invariant of* $B_1\backslash\{t_r\}$ *or* B_2.

Theorem 3.4 (*Property preservation under Transition Refinement* $B_1(t_r \to B_2)$). *The following propositions are valid under Transformation TR.*

(1) *If both* B_1 *and* B_2 *are SM* (*almost MG, FC nets, AC nets*), *so is* $B_1(t_r \to B_2)$.
(2) *If both* B_1 *and* B_2 *are conservative* (*structurally bounded*), *so is* $B_1(t_r \to B_2)$.
(3) *If* B_1 *is consistent* (*repetitive*) *and* B_2 *is almost consistent* (*almost repetitive*), *then* $B_1(t_r \to B_2)$ *is consistent* (*repetitive*).
(4) $\mathrm{Rank}(B_1(t_r \to B_2)) = \mathrm{Rank}(B_1) + \mathrm{Rank}(B_2) - h + d - 2$, *where h and d are computed according to Theorem* 3.3.
(5) $|C(B_1(t_r \to B_2))| = |C(B_1)| + |C(B_2)| - 2$.
(6) *If both* B_1 *and* B_2 *satisfy the RC-property, so does* $B_1(t_r \to B_2)$ *provided that* $d - h = 1$ *in Proposition* 4.
(7) *If* B_1 *has a minimal SM-cover and* B_2 *almost has a minimal SM-cover, then* $B_1(t_r \to B_2)$ *has a minimal SM-cover*.
(8) *Suppose D is a siphon of* B_1. *If* $r_o \notin D$ *or* ${}^\bullet p_x \subseteq p_e^\bullet$ *in* B_2, *then D is a siphon of* $B_1(t_r \to B_2)$. Suppose D is a siphon of B_2. If $p_e \notin D$ or ($p_x \in D$ and ${}^\bullet r_i \subseteq r_o^\bullet$ in B_1), then D is a siphon of $B_1(t_r \to B_2)$.
(9) *Suppose D is a trap of* B_1. If $r_i \notin D$ *or* $p_e^\bullet \subseteq {}^\bullet p_x$ *in* B_2, then *D is a trap of* $B_1(t_r \to B_2)$. *Suppose D is a trap of* B_2. If $p_x \notin D$ or ($p_e \in D$ *and* $r_o^\bullet \subseteq {}^\bullet r_i$ *in* B_1), *then D is a trap of* $B_1(t_r \to B_2)$.
(10) *If* B_1 *is also a Petri net process with or without the re-initiation path, then* $Z(B_1(t_r \to B_2)) = Z(B_1) + Z(B_2) - 1$.

(11) *Suppose* B_1 *is a Petri net process and* $\mathrm{LP}(B_1, t_\mathrm{r})$ *is the longest e-path within* B_1 *containing* t_r. *Then,* $\mathrm{LP}(B_1(t_\mathrm{r} \to B_2)) = \max\{\mathrm{LP}(B_1), \mathrm{LP}(B_1, t_\mathrm{r}) + \mathrm{LP}(B_2) - 1\}$.
(12) *If* (B_1, M_r) *is bounded by* k_1 *and* (B_2, M_e) *is bounded by* k_2, *then* $(B_1(t_\mathrm{r} \to B_2), M_\mathrm{i})$ *is bounded by max* $\{k_1, k_2\}$.
(13) *If* (B_1, M_r) *is live and* (B_2, M_e) *is almost live, then* $(B_1(t_\mathrm{r} \to B_2), M_\mathrm{i})$ *is live*.
(14) *If* (B_1, M_r) *is reversible and* (B_2, M_e) *is almost reversible, then* $(B_1(t_\mathrm{r} \to B_2), M_\mathrm{i})$ *is reversible*.

4. Property-preserving refinements of places

To refine a place, our approach is to first convert the place to a transition and then refine the transition by Transformation TR of Section 3. The conversion includes splitting the place into two places connected by a transition. Therefore, in order to obtain similar results as for transition refinement, we have to show that this conversion also preserves the same properties.

Transition PS (*Place splitting* $B \to B'$)(Fig. 2:). Let (B, M_r) be a marked net, where $B = (N, p_\mathrm{r})$ and $N = (P, T, F)$. Splitting place p_r of (B, M_r) results in a marked net (B', M'_r), where

- $B' = (N', r_\mathrm{i}, r_\mathrm{o}, t_\mathrm{r})$; $N' = (P', T', F')$.
- $P' = P \cup \{r_\mathrm{i}, r_\mathrm{o}\} - \{p_\mathrm{r}\}$; $T' = T \cup \{t_\mathrm{r}\}$; $F' = F \cup \{(r_\mathrm{i}, t_\mathrm{r}), (t_\mathrm{r}, r_\mathrm{o})\} \cup \{(x, r_\mathrm{i}) | x \in {}^\bullet p_\mathrm{r}\} \cup \{(r_\mathrm{o}, x) | x \in p_\mathrm{r}^\bullet\} - \{(x, p_\mathrm{r}) | x \in {}^\bullet p_\mathrm{r}\} - \{(p_\mathrm{r}, x) | x \in p_\mathrm{r}^\bullet\}$.
- $M'_\mathrm{r}(r_\mathrm{i}) = M_\mathrm{r}(p_\mathrm{r})$, $M'_\mathrm{r}(r_\mathrm{o}) = 0$ and $M'_\mathrm{r}(p) = M_\mathrm{r}(p)$ for $p \in P - \{p_\mathrm{r}\}$.

Theorem 4.1 (*Property preservation under Transformation PS*). *Suppose* (B', M'_r) *is obtained from* (B, M_r) *by Transformation PS. Then, the following propositions are valid*.

(1) *If B is an SM* (*MG, FC net, AC net*), *so is* B'.
(2) *If B is conservative* (*structurally bounded*), *so is* B'.
(3) *If B is consistent* (*repetitive*), *so is* B'.
(4) $\mathrm{Rank}(B') = \mathrm{Rank}(B) + 1$.
(5) $|C(B')| = |C(B)| + 1$.
(6) *If B satisfies the RC-property, so does* B'.
(7) *If B has a minimal SM-cover, so does* B'.
(8) *Suppose D is a siphon of B. If* $p_\mathrm{r} \notin D$, *then D is a siphon of* B'. *If* $p_\mathrm{r} \in D$, *then* $D - \{p_\mathrm{r}\} \cup \{r_\mathrm{i}, r_\mathrm{o}\}$ *is a siphon of* B'.
(9) *Suppose D is a trap of B. If* $p_\mathrm{r} \notin D$, *then D is a trap of* B'. *If* $p_\mathrm{r} \in D$, *then* $D - \{p_\mathrm{r}\} \cup \{r_\mathrm{i}, r_\mathrm{o}\}$ *is a trap of* B'.
(10) *Suppose B is a Petri net process. Then,* $Z(B') = Z(B)$.
(11) *Suppose B is a Petri net process. Let* $\mathrm{LP}(B, p_\mathrm{r})$ *be the longest e-path within B containing* p_r. *Then,* $\mathrm{LP}(B') = \max\{\mathrm{LP}(B), \mathrm{LP}(B, p_\mathrm{r}) + 1\}$.

(12)–(14) *If* (B, M_r) *is bounded* (*live, reversible*), *so is* (B', M'_r).

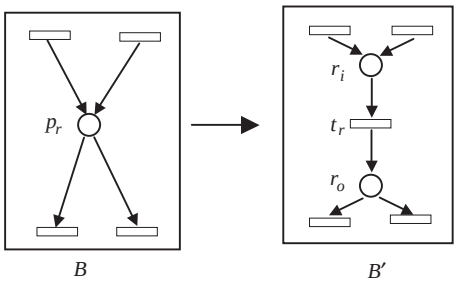

Fig. 2. Transformation PS.

Transformation PR (*Place Refinement* $B_1(p_\mathrm{r} \to B_2)$). Let (B_1, M_r) be a marked net and (B_2, M_c) be a Petri net process. Place p_r of (B_1, M_r) can be replaced with (B_2, M_c) by the following two steps, resulting in a marked net (B, M_i):

(1) $B_1 \to B'$ (by Transformation PS), creating a new transition t_r
(2) $B = B_1(p_\mathrm{r} \to B_2)) = B'(t_\mathrm{r} \to B_2)$ (by Transformation TR).

Theorem 4.2 (*Property preservation under Place Refinement* $B_1(p_\mathrm{r} \to B_2)$). *Let B_1 be a pure ordinary Petri net, B_2 be a Petri net process and $B_1(p_\mathrm{r} \to B_2)$ be obtained from B_1 and B_2 by Transformation PR. Then, the following propositions are valid*:

(1) *If both B_1 and B_2 are SM* (*almost MG, FC nets, AC nets*), *so is $B_1(p_\mathrm{r} \to B_2)$*.
(2) *If both B_1 and B_2 are conservative* (*structurally bounded*), *so is $B_1(p_\mathrm{r} \to B_2)$*.
(3) *If B_1 is consistent* (*repetitive*) *and B_2 is almost consistent* (*almost repetitive*), *then $B_1(p_\mathrm{r} \to B_2)$ is consistent* (*repetitive*).
(4) $\mathrm{Rank}(B_1(p_\mathrm{r} \to B_2)) = \mathrm{Rank}(B_1) + \mathrm{Rank}(B_2) - h + d - 1$, *where h and d are computed according to Theorem* 3.3 (*with B_1 replaced by B', the result of Transformation PS*).
(5) $|C(B_1(p_\mathrm{r} \to B_2))| = |C(B_1)| + |C(B_2)| - 1$.
(6) *If both B_1 and B_2 satisfy the RC-property, so does $B_1(p_\mathrm{r} \to B_2)$ provided that $d - h = 1$ in Proposition* 4.
(7) *If B_1 has a minimal SM-cover and B_2 almost has a minimal SM-cover, then $B_1(p_\mathrm{r} \to B_2)$ has a minimal SM-cover*.
(8) *Suppose D is a siphon of B_1. If $p_\mathrm{r} \notin D$, then D is a siphon of $B_1(p_\mathrm{r} \to B_2)$. If $p_\mathrm{r} \in D$ and ${}^\bullet p_\mathrm{x} \subseteq p_\mathrm{e}^\bullet$ in B_2, then $D - \{p_\mathrm{r}\} \cup \{p_\mathrm{i}, p_\mathrm{o}\}$ is a siphon of $B_1(p_\mathrm{r} \to B_2)$. Suppose D is a siphon of B_2. If $p_\mathrm{e} \notin D$, then D is a siphon of $B_1(p_\mathrm{r} \to B_2)$. If $\{p_\mathrm{e}, p_\mathrm{x}\} \subseteq D$ and ${}^\bullet p_\mathrm{r} \subseteq p_\mathrm{r}^\bullet$ in B_1, then $D - \{p_\mathrm{e}, p_\mathrm{x}\} \cup \{p_\mathrm{i}, p_\mathrm{o}\}$ is a siphon of $B_1(p_\mathrm{r} \to B_2)$*.
(9) *Suppose D is a trap of B_1. If $p_\mathrm{r} \notin D$, then D is a trap of $B_1(p_\mathrm{r} \to B_2)$. If $p_\mathrm{r} \in D$ and $p_\mathrm{e}^\bullet \subseteq {}^\bullet p_\mathrm{x}$ in B_2, then $D - \{p_\mathrm{r}\} \cup \{p_\mathrm{i}, p_\mathrm{o}\}$ is a trap of $B_1(p_\mathrm{r} \to B_2)$. Suppose D is a trap of B_2. If $p_\mathrm{x} \notin D$, then D is a trap of $B_1(p_\mathrm{r} \to B_2)$. If $p_\mathrm{e}, p_\mathrm{x} \in D$ and $p_\mathrm{r}^\bullet \subseteq {}^\bullet p_\mathrm{r}$ in B_1, then, $D - \{p_\mathrm{e}, p_\mathrm{x}\} \cup \{p_\mathrm{i}, p_\mathrm{o}\}$ is a trap of $B_1(p_\mathrm{r} \to B_2)$*.
(10) *Suppose B_1 is also a Petri net process with or without the re-initiation path. Then, $Z(B_1(p_\mathrm{r} \to B_2)) = Z(B_1) + Z(B_2) - 1$*.

(11) *Suppose* B_1 *is also a Petri net process. Let* $\mathrm{LP}(B_1, p_\mathrm{r})$ *be the longest path within* B_1 *containing* p_r. *Then,* $\mathrm{LP}(B_1(p_\mathrm{r} \to B_2)) = \max\{\mathrm{LP}(B_1), \mathrm{LP}(B_1, p_\mathrm{r}) + \mathrm{LP}(B_2)\}$.

(12) *If* (B_1, M_r) *is bounded by* k_1 *and* (B_2, M_e) *is bounded by* k_2, *then* $(B_1(p_\mathrm{r} \to B_2), M_\mathrm{i})$ *is bounded by* $\max(k_1, k_2)$.

(13) *If* (B_1, M_r) *is live,* p_r *has at least one input transition and* (B_2, M_e) *is almost live, then* $(B_1(p_\mathrm{r} \to B_2), M_\mathrm{i})$ *is live.*

(14) *If* (B_1, M_r) *is reversible and* (B_2, M_e) *is almost reversible,* $(B_1(p_\mathrm{r} \to B_2), M_\mathrm{i})$ *is reversible.*

5. An example for illustrating transition refinement

Description of a complaint-processing workflow system (Fig. 3 and Table 1):

The workflow system (B, p_1) operates as follows: When a complaint is launched, it will be registered. Then, a questionnaire is sent to the complainant while initial evaluation of the complaint is commenced. The response will be processed if returned within 2 weeks. Otherwise, it is discarded. Based on the result of the initial evaluation, the complaint is either formally processed or ignored. The actual processing of the complaint is delayed until the questionnaire is processed or a time-out has occurred. Processing of the complaint is monitored until all issues have been resolved and reprocessing may be warranted if there are any unsolved issues. Finally, the complaint is archived together with the questionnaire.

Petri net specification of the complaint-processing workflow system (Figs. 4 and 5):

System $B = B_1(t_\mathrm{r} \to B_2)$ is obtained by Transformation TR, where

(a) B_1 specifies the main system, where t_r is the operation which triggers Process B_2

(b) B_2 specifies the process for handling the questionnaire.

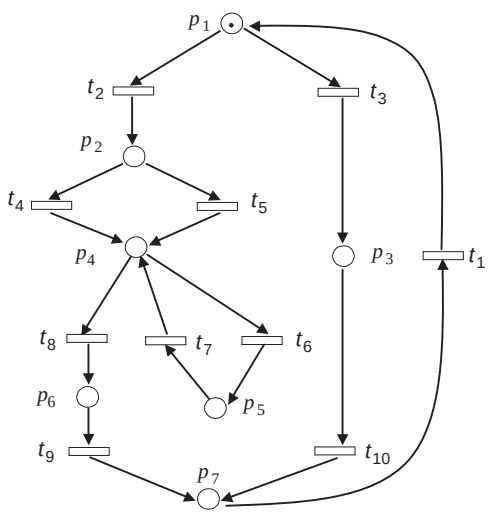

Fig. 3. Petri net representation of a workflow system B for processing complaints.

Table 1
Interpretation of places and transitions of the workflow system (Fig. 3)

Place and transition	Interpretation
p_1	Register a complaint
p_2	Waiting for the return of a questionnaire
p_3	The activity of normal processing of a complaint
p_4	Ready to process or finish the whole processing of a questionnaire
p_5	Ready to hold a questionnaire for further processing or reprocessing of a complaint
p_6	Finish the whole processing of a questionnaire
p_7	Archive a complaint together with a questionnaire
t_1	Register a new complaint
t_2	Send a questionnaire to a complainant
t_3	Evaluate a complaint
t_4	Discard a questionnaire not returned within 2 weeks
t_5	Process a questionnaire
t_6	Process a complaint
t_7	Hold the questionnaire for further processing or reprocessing of a complaint
t_8	Quit the activity of the questionnaire
t_9 and t_{10}	Finish the whole processing of the complaint

Properties of B_1 (matrix V_1 of B_1 is shown in Fig. 5):

(1) B_1 is SM, FC and AC but not MG because $|p_1^\bullet| > 1$ and $|{}^\bullet p_7| > 1$.

(2) B_1 is conservative and structurally bounded because $\alpha_1 \geqslant 1$ and $\alpha_1 V_1 = 0$, where $\alpha_1 = (x_5, x_4, \ldots, x_1) = (1, 1, 1, 1, 1)$.

(3) B_1 is consistent and repetitive because $\beta_1 \geqslant 1$ and $V_1\beta_1 = 0$, where $\beta_1 = (x_1, x_2, \ldots, x_6) = (1, 2, 1, 1, 1, 1)$.

(4) Since the first four rows of V_1 are linearly independent and the last row is the negative sum of the other rows, $\mathrm{Rank}(B_1) = 4$.

(5) $|C(B_1)| = 5$. The five clusters are: $\{p_1, t_2, t_3\}$, $\{p_2, t_\mathrm{r}\}$, $\{p_6, t_9\}$, $\{p_3, t_{10}\}$ and $\{p_7, t_1\}$.

(6) B_1 satisfies the RC-property because $\mathrm{Rank}(B_1) = |C(B_1)| - 1$.

(7) B_1 is SM-coverable with two SM-components: $p_1t_2p_2t_\mathrm{r}p_6t_9p_7t_1p_1$ and $p_1t_3p_3t_{10}p_7t_1p_1$.

(8) and (9) $D = \{p_1, p_2, p_3, p_6, p_7\}$ is both a siphon and a trap of B_1 because ${}^\bullet D = D^\bullet$.

(10) Since B_1 has a reinitiation path $p_7t_1p_1$, $z(B_1) = 2 - 1 + 1 = 2$. The two independent e-paths which cover all arcs of B_1 are: $p_1t_2p_2t_\mathrm{r}p_6t_9p_7$ and $p_1t_3p_3t_{10}p_7$.

(11) $\mathrm{LP}(B_1) = 3$. The longest e-path of B_1 from p_1 to p_7 is: $p_1t_2p_2t_\mathrm{r}p_6t_9p_7$.

(12)–(14) (B_1, p_1) is safe, live and reversible.

Properties of B_2 (matrices V_2 and $V_{2\mathrm{a}}$ of B_2 are shown in Fig. 5):

(1) B_2 is SM, FC and AC but is not an almost MG since $|p_2^\bullet| > 1$, $|{}^\bullet p_4| > 1$ and $|p_4^\bullet| > 1$.

(2) B_2 is conservative and structurally bounded because $\alpha_2 \geqslant 1$ and $\alpha_2 V_2 = 0$, where $\alpha_2 = (y_1, y_2, y_3, y_4) = (1, 1, 1, 1)$.

(3) B_2 is almost consistent and almost repetitive because $\beta_2 \geqslant 1$ and $V_{2\mathrm{a}}\beta_2 = 0$, where $\beta_2 = (y_1, y_2, y_3, y_4, y_5, z) = (1, 1, 1, 1, 2, 2)$.

(4) Since columns t_4, t_6 and t_8 are linearly independent, $\mathrm{Rank}(B_2) = 3$.
(5) B_2 has four clusters: $\{p_2, t_4, t_5\}, \{p_4, t_6, t_8\}, \{p_5, t_7\}$ and $\{p_6\}$. Hence, $|C(B_2)| = 4$.
(6) B_2 satisfies the RC-property because $\mathrm{Rank}(B_2) = |C(B_2)| - 1$.
(7) B_2 is almost SM-coverable with $B_{2\mathrm{a}}$ as the only SM-component.
(8) Examples of siphons: $\{p_2, p_4, p_5\}, \{p_2, p_4, p_5, p_6\}$.
(9) Examples of traps: $\{p_4, p_5, p_6\}, \{p_2, p_4, p_5, p_6\}$.
(10) $Z(B_2) = 3$. The three independent e-paths which cover all arcs are: $p_2t_4p_4t_8p_6$, $p_2t_5p_4t_8p_6$ and $p_2t_5p_4t_6p_5t_7p_4t_8p_6$.
(11) $\mathrm{LP}(B_2) = 4$. The longest e-path is $p_2t_5p_4t_6p_5t_7p_4t_8p_6$.
(12)–(14) (B_2, p_2) is bounded, almost live and reversible.

Properties of B: The results listed below follow from Theorem 3.4 and V (Fig. 6).

(1) B is an SM, not an almost MG, an FC net and an AC net.
(2) B is conservative and structurally bounded because $\alpha \geqslant 1$ and $\alpha V = 0$, where $\alpha = (y_1\alpha_1[P_1 - \{p_2, p_4\}], x_2y_1, x_1y_2, x_1\alpha_2[P_2 - \{p_2, p_4\}]) = (1, 1, 1, 1, 1, 1, 1)$. (Note: α_1 and α_2 are taken from the descriptions of B_1 and B_2 given above, respectively.)
(3) B is almost consistent and repetitive because $\exists\beta = (z\beta_1[T_1 - \{t_\mathrm{r}\}],\ x_1\beta_2[T_2]) = (zx_2, \ldots, zx_6, x_1y_1, \ldots, x_1y_5) = (4, 2, 2, 2, 2, 1, 1, 1, 1, 2) \geqslant 1$, such that $V\beta = 0$. (Note: β_1 and β_2 are taken from the descriptions of B_1 and B_2 given above, respectively.)
(4) In B_1, $t_\mathrm{r} = (0, 0, 0, -1, 1)$. t_r occurs in the T-invariant: $t_\mathrm{r} + t_1 + t_2 + t_9 = 0$. Hence, $h = 0$. Also, $B_1\backslash\{t_\mathrm{r}\}$ has the unique S-invariant: $p_1 + p_3 + p_7 + p_2 + p_6 = 0$ and B_2 has the unique S-invariant: $p_2 + p_6 + p_4 + p_5 = 0$. This satisfies Condition (3) in Case 2 of Theorem 3.3. Hence, $d = 1$. It follows from Theorem 3.3 that $\mathrm{Rank}(B) = \mathrm{Rank}(B_1) + \mathrm{Rank}(B_2) - h + d - 2 = 4 + 3 - 1 = 6$.
(5) $|C(B)| = |C(B_1)| + |C(B_2)| - 2 = 5 + 4 - 2 = 7$. The 7 clusters are: $\{p_1, t_2, t_3\}$, $\{p_2, t_4, t_5\}, \{p_4, t_6, t_8\}$, $\{p_5, t_7\}$, $\{p_6, t_9\}$, $\{p_3, t_{10}\}$ and $\{p_7, t_1\}$.
(6) Since both B_1 and B_2 satisfy the RC-property and $d - h = 1$, B satisfies the RC-property. (This result is consistent with Propositions 4 and 5.)
(7) B is SM-coverable. The two SM-components of B are shown in Fig. 7.
(8) and (9) Siphon (trap) $D_1 = \{p_1, p_2, p_6, p_3, p_7\}$ of B_1 is not a siphon (trap) of B because it does not satisfy the condition of Theorem 3.4. Similarly, siphons $D_2 = \{p_2, p_4, p_5\}$ and $D_3 = \{p_2, p_4, p_5, p_6\}$ of B_2 are not siphons of B. D_3 is not a trap of B either.
(10) $Z(B) = Z(B_1) + Z(B_2) - 1 = 2 + 3 - 1 = 4$. By applying the cyclomatic complexity formula directly, $Z(B) = 20 - (7 + 10) + 1 = 4$. The four linearly independent e-paths which cover all arcs except (p_7, t_1) and (t_1, p_1) are: $p_1t_2p_2t_4p_4t_8p_6t_9p_7$, $p_1t_2p_2t_5p_4t_8p_6t_9p_7$, $p_1t_2p_2t_5p_4t_6p_5t_7p_4t_8p_6t_9p_7$ and $p_1t_3p_3t_{10}p_7$.
(11) $\mathrm{LP}(B) = \max\{\mathrm{LP}(B_1\backslash\{t_1\}), \mathrm{LP}(B_1\backslash\{t_1\}, t_\mathrm{r}) + \mathrm{LP}(B_2) - 1\} = \max\{2, 3 + 4 - 1\} = 6$. The longest e-path for one iteration of $B\backslash\{t_1\}$ is: $p_1t_2p_2t_5p_4t_6p_5t_7p_4t_8p_6t_9p_7$.
(12)–(14) (B, p_1) is bounded, live and reversible.

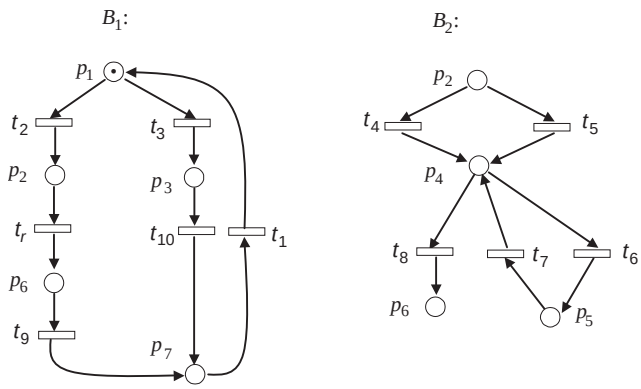

Fig. 4. Basic component processes B_1 and B_2 for B of Fig. 3 (B_1 has a re-initiation path and B_2 does not.).

$$
\begin{array}{c}
 \\ p_1 \\ p_3 \\ p_7 \\ p_2 \\ p_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} t_{\mathrm{r}} & t_1 & t_3 & t_{10} & t_2 & t_9 \end{array} \\
\begin{pmatrix}
0 & 1 & -1 & 0 & -1 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 1 \\
-1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}
\end{array}
\qquad
\begin{array}{c}
 \\ p_2 \\ p_6 \\ p_4 \\ p_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} t_6 & t_7 & t_4 & t_5 & t_8 & t_{2a} \end{array} \\
\begin{pmatrix}
0 & 0 & -1 & -1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 0 \\
1 & -1 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Fig. 5. Incidence matrices V_1 for B_1 and $V_{2\mathrm{a}}$ for $B_{2\mathrm{a}}$ (V_2 for B_2 is the part of V_{2a} without column $t_{2\mathrm{a}}$).

$$
\begin{array}{c}
 \\ p_1 \\ p_3 \\ p_7 \\ p_2 \\ p_6 \\ p_4 \\ p_5
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} t_1 & t_3 & t_{10} & t_2 & t_9 & t_6 & t_7 & t_4 & t_5 & t_8 \end{array} \\
\begin{pmatrix}
1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Fig. 6. Incidence matrix V for B (column t_{r} is not included).

6. Conclusion

In the modular approach for designing complex systems that have many desirable properties, current research aims at finding transformations that can preserve as many properties as possible. For Petri nets, transition and place refinements are important transformations for building complex systems from simple components. Previous study on refinement techniques [16,17] focused mainly on their preservation of behavioral properties such as liveness and boundedness. Based on a more general version of transition and place refinements, where the refinement net simulates a software process, this paper proposes various conditions under which nineteen properties will be preserved. These nineteen properties can accommodate the design of complex systems and most of them are structural. Our re-

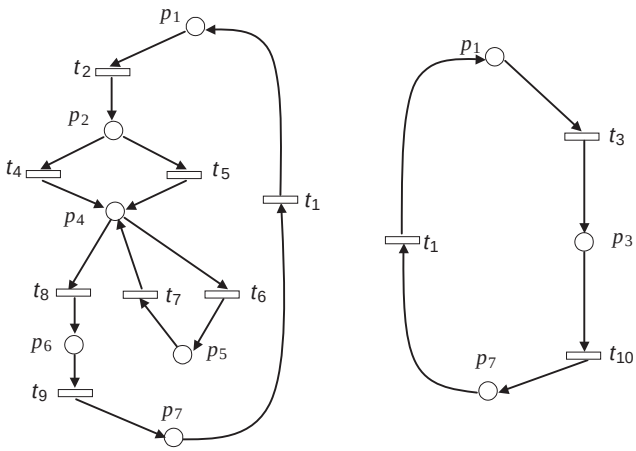

Fig. 7. Two SM-components of B.

sults extend capability of refinements from the preservation of behavioral properties to the preservation of structural properties. They also enhance the property-preserving approach and characterization-based approach for system verification.

Acknowledgements

This research was financially supported under Grant CityU1072/00E from Research Grant Council, the Government of Hong Kong.

Appendix

Proof for Lemma 3.1. This follows from the assumption of proper termination on (B_2, M_c).

Proof for Lemma 3.2 (*By mathematical induction on* σ). For $\sigma = \lambda$, Lemma 3.2 obviously holds because $\#(\lambda, {}^\bullet p_o) = \#(\lambda, p_i^\bullet) = 0$. Suppose $\forall \sigma' \in L(B, M_i)$ where $|\sigma'| \leqslant n$, $\#(\sigma', {}^\bullet p_o) + 1 \geqslant \#(\sigma', p_i^\bullet) \geqslant \#(\sigma', {}^\bullet p_o)$. Let $\sigma = \sigma' t$. For $t \notin p_i^\bullet \cup {}^\bullet p_o$, $\#(\sigma, p_i^\bullet) = \#(\sigma', p_i^\bullet)$ and $\#(\sigma, {}^\bullet p_o) = \#(\sigma', {}^\bullet p_o)$. Hence, $\#(\sigma, {}^\bullet p_o) + 1 \geqslant \#(\sigma, p_i^\bullet) \geqslant \#(\sigma, {}^\bullet p_o)$. For $t \in p_i^\bullet$, since B_2 initiates and terminates properly, we have $\#(\sigma, p_i^\bullet) = \#(\sigma', p_i^\bullet) + 1$ and $\#(\sigma, {}^\bullet p_o) = \#(\sigma', {}^\bullet p_o)$. Hence, $\#(\sigma, {}^\bullet p_o) + 1 \geqslant \#(\sigma, p_i^\bullet) \geqslant \#(\sigma, {}^\bullet p_o)$. For $t \in {}^\bullet p_o$, the proof is similar.

Proof for Theorem 3.1 (*By mathematical induction on the length of* σ). For $\sigma = \lambda$, obviously $M = M_i$. By Definition 3.2, $f_1(\sigma) = f_2(\sigma) = \lambda$. Hence $f_1(\sigma) \in L(B_1, M_r)$ and

$f_2(\sigma) \in L(B_2, M_e)$. Since $M_i = M_r + M_c$ as defined in Transformation TR, $M_r = M_1$ and $M_e = M_2$. Since obviously $M_r[B_1, \lambda\rangle M_r$ and $M_e[B_2, \lambda\rangle M_e$, it follows that $M_r[B_1, f_1(\sigma)\rangle M_1$ and $M_e[B_2, f_2(\sigma)\rangle M_2$. Next, suppose Propositions 1 and 2 hold for all μ where $|\mu| \leqslant n$. That is, $M_i[B, \mu\rangle M'$ implies:

(1) $f_1(\mu) \in L(B_1, M_r)$; $f_2(\mu) \in L(B_2, M_e)$.

(2) $M_r[B_1, f_1(\mu)\rangle M_1'$, where $M_1'(r_i) = M'(p_i) + (\#(\mu, p_i^\bullet) - \#(\mu, {}^\bullet p_o))$, $M_1'(r_o) = M'(p_o)$ and $M_1'(p) = M'(p)$ for $p \in P_1 - \{r_i, r_o\}$; $M_e[B_2, f_2(\mu)\rangle M_2'$, where $M_2'(p_e) = M'(p_i)$, $M_2'(p_x) = 0$, $M_2'(p) = M'(p)$ for $p \in P_2 - \{p_e, p_x\}$.

For $\sigma = \mu t \in L(B, M_i)$, i.e., $M_i[B, \mu\rangle M'[B, t\rangle M$, proof of Propositions 1 and 2 for σ proceeds as follows:

(1) For Proposition 1, three cases should be considered:

(a) If $t \in T_2 - {}^\bullet p_o$, $f_1(\mu) = f_1(\mu) \in L(B_1, M_r)$, $f_2(\sigma) = f_2(\mu)t$. By Definition 3.2, $M_2'(p_e) = M'(p_i)$, $M_2'(p_x) = 0$ and $M_2'(p) = M'(p)$ for $p \in P_2 - \{p_e, p_x\}$. Since p_x is not in ${}^\bullet t$ and t is enabled at M' in B, t is also enabled at M_2' in B_2. Hence, $f_2(\sigma) \in L(B_2, M_e)$.

(b) If $t \in T_1 - \{t_r\}$, $f_1(\sigma) = f_1(\mu)t$ and $f_2(\sigma) = f_2(\mu) \in L(B_2, M_e)$. Since $M_1' \geqslant M'$ by Definition 3.2 and the fact that t is enabled at M' in B, t is also enabled at M_1' in B_1. Hence, $f_1(\sigma) \in L(B_1, M_r)$.

(c) If $t \in {}^\bullet p_o$, $f_1(\sigma) = f_1(\mu)t_r$ and $f_2(\sigma) = \gamma \in L(B_2, M_e)$. Since μ is followed by $t \in {}^\bullet p_o$, Lemma 3.2 implies that $\#(\mu, p_i^\bullet) = \#(\mu, {}^\bullet p_o) + 1$. Hence, $M_1'(r_i) = M(p_i) + \#(\mu, p_i^\bullet) - \#(\mu, {}^\bullet p_o) \geqslant 1$ and t_r is enabled at M_1' in B_1. Hence, $f_1(\sigma) \in L(B_1, M_r)$.

(2) By Proposition 1, we can assume that $M'[B, t\rangle M$, $M_1'[B_1, f_1(t)\rangle M_1$ and $M_2'[B_2, f_2(t)\rangle M_2$. Proposition 2 will follow if M_1 and M_2 can be derived from M according to Definition 3.2. To show this, we consider the following five cases:

(a) If $t \in T_2 - p_e^\bullet - {}^\bullet p_x = T_2 - p_i^\bullet - {}^\bullet p_o$, then $\#(\sigma, p_i^\bullet) = \#(\mu, p_i^\bullet)$ and $\#(\sigma, {}^\bullet p_o) = \#(\mu, {}^\bullet p_o)$. Also, the token distribution within B_1 is not affected, i.e., $M_1(p) = M_1'(p) = M(p)\ \forall p \in P_1$. Within B_2, firing t does not affect the tokens in p_e and p_x. Hence, $M_2(p_e) = M_2'(p_e) = M'(p_i) = M(p_i)$ and $M_2(p_x) = M_2'(p_x) = 0$, $M_2(p) = M_2'(p) + w(t, p) - w(p, t) = M'(p) + w(t, p) - w(p, t) = M(p)$ for $p \in P_2 - \{p_e, p_x\}$.

(b) If $t \in T_1 - \{t_r\}$, then $\#(\sigma, p_i^\bullet) = \#(\mu, p_i^\bullet)$, $\#(\sigma, {}^\bullet p_o) = \#(\mu, {}^\bullet p_o)$. Within B_1, we have: $M_1(r_i) = M_1'(r_i) + \mathrm{w}(t, r_i) = M'(p_i) + (\#(\mu, p_i^\bullet) - \#(\mu, {}^\bullet p_o)) + w(t, r_i) = M(p_i) + \#(\sigma, p_i^\bullet) - \#(\sigma, {}^\bullet p_o)$, $M_1(r_o) = M_1'(r_o) - w(r_o, t) = M'(p_o) - w(r_o, t) = M(p_o)$ and $M_1(p) = M_1'(p) + w(t, p) - w(p, t) = M'(p) + w(t, p) - w(p, t) = M(p)$ for $p \in P_1 - \{r_i, r_o\}$. Within B_2, firing t does not affect the tokens distribution in B_2. Hence, $M_2(p_e) = M_2'(p_e) + w(t, p_e) = M'(p_i) + w(t, p_i) = M(p_i)$ and $M_2(p_x) = M_2'(p_x) = 0$, $M_2(p) = M_2'(p) = M(p)$ for $p \in P_2 - \{p_e, p_x\}$.

(c) If $t \in p_i^\bullet - {}^\bullet p_o$, then $\#(\sigma, p_i^\bullet) = \#(\mu, p_i^\bullet) + 1$, $\#(\sigma, {}^\bullet p_o) = \#(\mu, {}^\bullet p_o)$ and $f_1(t) = \lambda$. Within B_1, firing t does not affect the token distribution in B_1. $M(p) = M'(p)$ for $p \in P_1$ and $M(p_i) = M'(p_i) - 1$. Hence, $M_1(r_i) = M_1'(r_i) = M'(p_i) + \#(\mu, p_i^\bullet) - \#(\mu, {}^\bullet p_o) = M(p_i) + \#(\sigma, p_i^\bullet) - \#(\sigma, {}^\bullet p_o)$, $M_1(r_o) = M_1'(r_o) = M'(p_o) = M(p_o)$ and $M_1(p) = M_1'(p) = M'(p) = M(p)$ for $p \in P_1 - \{r_i, r_o\}$. Within B_2, it follows from Definitions 2.1 and 3.2 that $f_2(\mu) = \lambda$ or γ, and $f_2(\sigma) = t$. Hence, $M_2(p_e) = M_2'(p_e) - 1 = M'(p_e) - 1 = M(p_e)$, $M_2(p) = M_2'(p) + w(t, p) - w(p, t) =$

$M'(p)+w(t,p)-w(p,t)=M(p)$ for $p\in P_2-\{p_e,p_x\}$ and $M_2(p_x)=M_2'(p_x)=0$.

(d) If $t\in {}^\bullet p_o-p_i^\bullet$, then $\#(\sigma,p_i^\bullet)=\#(\mu,p_i^\bullet)$, $\#(\sigma,{}^\bullet p_o)=\#(\mu,{}^\bullet p_o)+1$ and $f_1(t)=t_r$. Within B_1, after firing t in B, $M(p)=M'(p)$ for $p\in P_1-\{r_o\}$, $M(p_i)=M'(p_i)$, $M(p_o)=M'(p_o)+1$ and $M_1(r_i)=M_1'(r_i)-1$. Hence, $M_1(r_i)=M_1'(r_i)-1=(M'(p_i)+\#(\mu,p_i^\bullet)-\#(\mu,{}^\bullet p_o))-1=M'(p_i)+\#(\sigma,p_i^\bullet)-\#(\sigma,{}^\bullet p_o)=M(p_i)+\#(\sigma,p_i^\bullet)-\#(\sigma,{}^\bullet p_o)$, $M_1(r_o)=M_1'(r_o)+1=M'(p_o)+1=M(p_o)$ and $M_1(p)=M_1'(p)=M'(p)=M(p)$ for $p\in P_1-\{r_i,r_o\}$. Within B_2, $f_2(\sigma)=\gamma$. $M_2(p_e)=M_2'(p_e)=M'(p_i)=M(p_i)$ and $M_2(p_x)=0$. By Definition 3.2, $M_2(p)=M_2'(p)+w(t,p)-w(p,t)=M'(p)+w(t,p)-w(p,t)=M(p)$ for $p\in P_2-\{p_e,p_x\}$.

(e) If $t\in p_i^\bullet\cap{}^\bullet p_o$, then $\#(\sigma,p_i^\bullet)=\#(\mu,p_o^\bullet)+1$, $\#(\sigma,{}^\bullet p_o)=\#(\mu,{}^\bullet p_o)+1$ and $f_1(t)=t_r$. Within B_1, after firing t in B, $M(p)=M'(p)$ for $p\in P_1-\{r_i,r_o\}$, $M(p_i)=M'(p_i)-1$, $M(p_o)=M'(p_o)+1$, $M_1(r_i)=M_1'(r_i)-1$ and $M_1(r_o)=M_1'(r_o)+1$. The rest of the proof is similar to Case d above. Within B_2, by Definitions 2.1 and 3.2, $f_2(\mu)=f_2(\sigma)=\gamma$. Hence, $M_2(p_e)=M_2'(p_e)-1=M'(p_i)-1=M(p_i)$, $M_2(p_x)=0$ and $M_2(p)=M_2'(p)+w(t,p)-w(p,t)=M'(p)+w(t,p)-w(p,t)=M(p)$ for $p\in P_2-\{p_e,p_x\}$.

Proof for Lemma 3.3. (1) It is obvious that if $M_e[B_2,\mu\rangle M_2$, then $\exists\sigma\in L(B,M_i)$ such that $(M_1+M_e)[B,\sigma\rangle(M_1+M_2)$, where μ is a subsequence of σ and M_1 is any marking of B_1. (2) Let $\gamma=t_i\ldots t_o$ be a firable e-sequence of B_2, where $t_i\in p_e^\bullet$ and $t_o\in{}^\bullet p_x$. Any $\mu\in L(B_1,M_r)$ can be expressed as $\mu_1\ldots t_r\ldots\mu_2\ldots t_r\mu_3\ldots$. It follows from Definition 3.2 that $\mu=f_1(\sigma)$. Next, we shall show that $\sigma\in L(B,M_i)$ by induction on μ. For $\mu=\lambda$, it is trivial that $\sigma=\lambda\in L(B,M_i)$. Suppose that, $\forall\mu'\in L(B_1,M_r)$ where $|\mu'|\leqslant n$, $\exists\sigma'\in L(B,M_i)$ such that $\mu'=f_1(\sigma')$. Let $M_1'=M_r[B_1,\mu'\rangle$ and $M'=M_i[B,\sigma'\rangle$. By Definition 3.2, $M'(p_i)=M_1'(r_i)-(\#(\sigma',p_i^\bullet)-\#(\sigma',{}^\bullet p_o))$ and $M'(p)=M_1'(p)$ for $p\in P_1-\{r_i\}$. By Lemma 3.2, $M'(p)\geqslant M_1(p)\ \forall p\in P_1$. For $\mu=\mu't\in L(B_1,M_r)$, consider two cases: (a) $t\in T_1-\{t_r\}$. Let $\sigma=\sigma't$. Since t is firable at M_1' in B_1, it is firable at M' in B. Hence, $\sigma\in L(B,M_i)$. (b) $t=t_r$. Let $\sigma=\sigma'\gamma$. By the result of Part (1), $\sigma\in L(B,M_i)$.

Proof for Theorem 3.2. Proposition 1 follows from the fact that B_2 initiates and terminates properly and that B_2 itself is initially marked with M_c which cannot initiate any firing in B_2. Proposition 2 follows from two facts: (a) B_2 is not re-enterable (i.e., B_2 cannot be initiated again unless its current execution cycle has terminated at M_x and the token deposited at p_x has been removed. This is guaranteed by the assumptions of proper initiation and proper termination on B_2. (b) The only initial marking that B_1 can create for B_2 is $M_e=p_e+M_c$. This is guaranteed by the assumptions that B_1 is not 2-enabled at t_r and that B_2 is non-reenterable. According to the discussions following Definition 3.2, from the viewpoint of B_1, a token deposited into r_i is held there during execution of B_2 and is removed only after B_2 has terminated. Hence, since B_1 is not 2-enabled at t_r, no more token can be put into r_i during the execution of B_2.

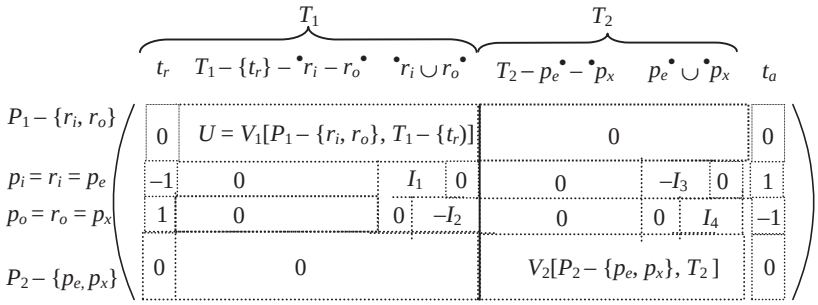

Fig. .8. Compositions of V_1, V_2, V_{2a} and V (see explanation below).

Proof for Theorem 3.3 (Fig. 8). (Note: The proof presented below is based on two facts: (1) Eliminations are always achieved by applying Gaussian row operations over V though their effects may be considered within individual submatrices. (2) If a non-isolated place (transition) is contained in an S-invariant (T-invariant), its corresponding row (column) in the incidence matrix can be expressed as a linear combination of the other rows (columns) of the invariant.) The conclusion of this theorem follows from two results: (a) $\mathrm{Rank}(B_1) = \mathrm{Rank}(B_1\backslash\{t_\mathrm{r}\}) + h$. (b) $\mathrm{Rank}(B_1(t_\mathrm{r} \to B_2)) = \mathrm{Rank}(B_1\backslash\{t_\mathrm{r}\}) + \mathrm{Rank}(B_2) + d - 2$.

(a) It is obvious that $\mathrm{Rank}(B_1) = \mathrm{Rank}(B_1\backslash\{t_\mathrm{r}\}) + h$, where $h = 0$ or 1. If t_r occurs in any T-invariant of B_1, column t_r is linearly dependent on the other columns and hence $h = 0$. Otherwise, column t_r is linearly independent and $h = 1$.

(b) Let V be divided into three horizontal blocks: BK_1 contains its top $|P_1| - 2$ rows, BK_2 its middle 2 rows, and BK_3 its bottom $|P_2| - 2$ rows. Consider the following three cases:

Case 1: $d = 2$. We shall prove for Condition (1) only. Proof for Condition (2) is similar. Consider four cases: (i) *Both S_1 and S_2 belong to $B_1\backslash\{t_\mathrm{r}\}$*. Within $B_1\backslash\{t_\mathrm{r}\}$, since $p_\mathrm{o} \in S_2$, row p_o can be expressed as a linear combination of rows of BK_1 and row p_i. It can thus be eliminated to zero. Also, since $p_\mathrm{i} \in S_1$ and $p_\mathrm{o} \notin S_1$, row p_i can be expressed as a linear combination of the rows of BK_1 alone and can be eliminated to zero. After these two eliminations, V becomes diagonal with diagonal blocks $BK_1[T_1 - \{t_\mathrm{r}\}]$ and B_2. Also, $\mathrm{Rank}(BK_1[T_1 - \{t_\mathrm{r}\}]) = \mathrm{Rank}(B_1\backslash\{t_\mathrm{r}\})$ and the rank of B_2 remains unchanged. Hence, $\mathrm{Rank}(B) = \mathrm{Rank}(BK_1[T_1 - \{t_\mathrm{r}\}]) + \mathrm{Rank}(B_2) = \mathrm{Rank}(B_1\backslash\{t_\mathrm{r}\}) + \mathrm{Rank}(B_2)$. (ii) *$S_1$ and S_2 both belong to B_2*. Proof is similar as (i). (iii) *S_2 belongs to $B_1\backslash\{t_\mathrm{r}\}$ whereas S_1 belongs to B_2*. Without affecting the ranks of $B_1\backslash\{t_\mathrm{r}\}$, B_2 and V, row p_o can be eliminated to zero within $B_1\backslash\{t_\mathrm{r}\}$ and row p_i to zero within B_2 (by means of the rows of BK_3 alone). Again, V becomes diagonal and hence $\mathrm{Rank}(B) = \mathrm{Rank}(B_1\backslash\{t_\mathrm{r}\}) + \mathrm{Rank}(B_2)$. (iv) *$S_1$ belongs to $B_1\backslash\{t_\mathrm{r}\}$ whereas S_2 belongs to B_2*. Proof is similar as (iii).

Case 2: $d = 1$. Without loss of generality, suppose that S_1 belongs to $B_1\backslash\{t_\mathrm{r}\}$ and that $p_\mathrm{i} \in S_1$. Then, within $B_1\backslash\{t_\mathrm{r}\}$, row p_i is linearly dependent on the other rows and can thus be eliminated to zero without altering the ranks of $B_1\backslash\{t_\mathrm{r}\}$, B_2 as well as V. On the other hand, for the reasons stated in Remark below, row p_o is linearly independent of the other rows within $B_1\backslash\{t_\mathrm{r}\}$ and B_2 (and thus within V as well). Hence, by ignoring row

p_o, V becomes diagonal and $\text{Rank}(B) = 1 + (\text{Rank}(B_1\backslash\{t_r\}) - 1) + (\text{Rank}(B_2) - 1) = \text{Rank}(B_1\backslash\{t_r\}) + \text{Rank}(B_2) - 1)$. (Remark: For Condition (1), since p_i already appears in S_1, p_o cannot appear in any S-invariant and hence cannot be dependent on the other rows within both $B_1\backslash\{t_r\}$ and B_2. For Condition (2), after p_i has been eliminated to zero within $B_1\backslash\{t_r\}$, row p_o cannot be linearly dependent on the other rows within $B_1\backslash\{t_r\}$, because, otherwise, $\exists S'$ of $B_1\backslash\{t_r\}$ such that $p_o \in S'$ but $p_i \notin S'$—contrary to Condition (2). Row p_o cannot be linearly dependent on the other rows within B_2 either because p_o does not belong to any S-invariant of B_2. Under Condition (3), the fact that row p_o cannot be linearly dependent on the other rows within $B_1\backslash\{t_r\}$ follows from the same argument as for Condition (2). Within B_2 it is due to the following argument: Within $B_1\backslash\{t_r\}$, in order to eliminate row p_i to zero by means of the S-invariant $a_1p_i + b_1p_o + c_1 = 0$, a multiple of (b_1/a_1) of row p_o (and some combination of other rows in BK_1) has to be added to row p_i. Since this is done over V, row p_i within B_2 becomes $p_i + (b_1/a_1)p_o$. Now, suppose row p_o is linearly dependent on the other rows within B_2, that is, $\exists a' \neq 0$ and $b' \neq 0$: $a'(p_i + (b_1/a_1)p_o) + b'p_o + c' = a'p_i + (a'b_1/a_1 + b')p_o + c' = 0$, where c' is a linear combination of the rows of BK_3. Then, since $a_2p_i + b_2p_o + c_2 = 0$ is the unique S-invariant of B_2 containing p_o, the coefficients of these two equations must be proportional. That is, $a'/a_2 = (a'b_1/a_1 + b')/b_2$ or $b'/a' = b_2/a_2 - b_1/a_1$. Since $b_2/a_2 = b_1/a_1, b' = 0$—a contradiction.)

Case 3: $d = 0$. Rows p_i and p_o are each linearly independent of the other rows within both $B_1\backslash\{t_r\}$ and B_2 (and hence also within V) because, otherwise, p_i or p_o will appear in at least one S-invariant. Then, after ignoring rows p_i and p_o, the ranks of $B_1\backslash\{t_r\}$, B_2 and V will each be reduced by 2 and V becomes diagonal. Hence, $\text{Rank}(B) = 2 + (\text{Rank}(B_1\backslash\{t_r\}) - 2) + (\text{Rank}(B_2) - 2) = \text{Rank}(B_1\backslash\{t_r\}) + \text{Rank}(B_2) - 2$.

Explanations for Fig. .8:

- B_1 denotes a refined net, B_2 a Petri net process and $B = B_1(t_r \to B_2)$ the result of applying Transformation TR to replace transition t_r of B_1 with B_2.
- $V(P, T) = V[P_1 \cup P_2, T_1 \cup T_2 - \{t_r\}]$ is the incidence matrix of B.
- $V_1(P_1, T_1)$ of B_1 occupies the first $|P_1|$ rows and first $|T_1|$ columns.
- $V_2(P_2, T_2)$ of B_2 occupies the bottom $|P_2|$ rows and the columns under T_2.
- $V_{2a}(P_2, T_2 \cup \{t_a\})$ occupies the bottom $|P_2|$ rows and the rightmost $|T_2| + 1$ columns.
- Row p_i has values: $p_i(t) = 1$ if $t \in {}^\bullet r_i$, $p_i(t) = -1$ if $t \in \{t_r\} \cup p_e{}^\bullet$ and $p_i(t) = 0$ if $t \in T_1 \cup T_2 - \{t_r\} - {}^\bullet r_i - p_e{}^\bullet$; and the row p_o has values: $p_o(t) = -1$ if $t \in r_o{}^\bullet$, $p_o(t) = 1$ if $t \in \{t_r\} \cup {}^\bullet p_x$, and $p_i(t) = 0$ if $t \in T_1 \cup T_2 - \{t_r\} - {}^\bullet p_x - r_o{}^\bullet$.

Proof for Theorem 3.4 (Fig. 8). For brevity, throughout the proof below, B denotes $B_1(t_r \to B_2)$.

(1) According to its construction, B satisfies the definition of SM (resp., almost MG, FC and AC) if both B_1 and B_2 are SM (resp., almost MG, FC and AC nets).

(2) Since B_1 and B_2 are conservative, $\exists \alpha_1 = (x_{|p1|}, \ldots, x_2, x_1) \geqslant 1$ and $\alpha_2 = (y_1, y_2, \ldots, y_{|P2|}) \geqslant 1$ such that $\alpha_1 V_1 = \alpha_2 V_2 = 0$. In particular, the first column of $\alpha_1 V_1 = 0$, i.e., $\alpha_1 V_1[P_1, \{t_r\}] = 0$, leads to $x_1 = x_2$. By Lemma 3.1, $M_e[B_2, *\rangle M_x$. It follows from AC-1 that $\alpha_2(p_e + M_c) = \alpha_2(p_x + M_c)$ and $y_1 = y_2$. Next, let $\alpha = (y_1\alpha_1[P_1 - \{r_i, r_o\}], x_2y_1, x_1y_2, x_1\alpha_2[P_2 - \{p_e, p_x\}]) = (x_{|p1|}y_1, \ldots, x_3y_1, x_2y_1, x_1y_2, x_1y_3, \ldots,$

$x_1 y_{|P2|}$). Then, $\alpha \geqslant 1$ and $\alpha V = (-x_2 y_1 + x_1 y_2, y_1 \alpha_1 [P_1 - \{r_\mathrm{i}, r_\mathrm{o}\}] V_1 [P_1 - \{r_\mathrm{i}, r_\mathrm{o}\}, T_1 - \{t_\mathrm{r}\}] + x_2 y_1 V_1 [r_\mathrm{i}, T_1 - \{t_\mathrm{r}\}] + x_1 y_2 V_1 [r_\mathrm{o}, T_1 - \{t_\mathrm{r}\}], x_2 y_1 V_2 [p_\mathrm{e}, T_2] + x_1 y_2 V_2 [p_\mathrm{x}, T_2] + x_1 \alpha_2 [P_2 - \{p_\mathrm{e}, p_\mathrm{x}\}] V_2 [P_2 - \{p_\mathrm{e}, p_\mathrm{x}\}, T_2]) = (0, y_1 \alpha_1 V_1 [P_1, T_1 - \{t_\mathrm{r}\}], x_1 \alpha_2 V_2) = 0$. That is, B is conservative. Similar argument as above shows that $\alpha_1 V_1 \leqslant 0$ and $\alpha_2 V_{2\mathrm{a}} \leqslant 0$ lead to $\alpha V \leqslant 0$. It follows that B is structurally bounded.

(3) Since B_1 is consistent, $\exists \beta_1 = (x_1, \ldots, x_{|T1|}) \geqslant 1$ such that $V_1 \beta_1 = 0$. Since B_2 is almost consistent, $\exists \beta_2 = (y_1, \ldots, y_{|T2|}, z) \geqslant 1$ such that $V_{2\mathrm{a}} \beta_2 = 0$. Let $\beta = (z\beta_1 [T_1 - \{t_\mathrm{r}\}], x_1 \beta_2 [T_2]) = (z x_2, \ldots, z x_{|T1|}, x_1 y_1, \ldots, x_1 y_{|T2|})$. Then, $\beta \geqslant 1$ and, since $V[P, t_\mathrm{r}] + V[P, t_\mathrm{a}] = 0$, we have $V\beta = (z V_1 [P_1, T_1 - \{t_\mathrm{r}\}] \beta_1 [T_1 - \{t_\mathrm{r}\}], 0) + (0, x_1 V_2 \beta_2 [T_2]) = ((z V_1 \beta_1, 0) - z x_1 V[P, t_\mathrm{r}]) + ((0, x_1 V_{2\mathrm{a}} \beta_2) - x_1 z V[P, t_\mathrm{a}]) = 0$. That is, B is consistent. Similar argument as above shows that $V_1 \beta_1 \geqslant 0$ and $V_{2\mathrm{a}} \beta_2 \geqslant 0$ lead to $V\beta \geqslant 0$. It follows that B is repetitive.

(4) Refer to Theorem 3.3.

(5) In forming B, since t_r is deleted, cluster $\{r_\mathrm{i}, t_\mathrm{r}\}$ of B_1 is destroyed and r_i is absorbed into the cluster $[p_\mathrm{e}]$ of B_2 to form the new cluster $[p_\mathrm{i}]$ of B. Also, the clusters $[r_\mathrm{o}]$ of B_1 and $[p_\mathrm{x}]$ of B_2 are merged to form the new cluster $[p_\mathrm{o}]$ of B. Also, no other clusters are created or destroyed. Hence, $|C(B)| = |C(B_1)| + |C(B_2)| - 2$.

(6) From Propositions 4 and 5, if $d - h = 1$, $\mathrm{Rank}(B) = \mathrm{Rank}(B_1) + \mathrm{Rank}(B_2) - 1 = (|C(B_1)| - 1) + (|C(B_2)| - 1) - 1 = |C(B)| - 1$.

(7) Let K_1 and K_2 be the minimal SM-covers of B_1 and $B_{2\mathrm{a}}$, respectively. Any SM-component in K_1 containing r_o must also contain r_i, t_r, and a directed path from r_o to r_i. Similarly, any SM-component in K_2 containing p_x must also contain p_e, t_a and a directed path from p_x to p_e. A minimal SM-cover for B can be created as follows. Let $H_1 = \{S \in K_1 | \{r_\mathrm{i}, t_\mathrm{r}, r_\mathrm{o}\} \subseteq S\}$ and $H_2 = \{S \in K_2 | \{p_\mathrm{e}, t_\mathrm{a}, p_\mathrm{x}\} \subseteq S\}$. Create S' by merging all SM-components of H_1 with all SM-components of H_2 at their common places p_i and p_o (i.e., fuse r_i with p_e and r_o with p_x and delete all t_r and t_a). Obviously, S' is an SM-component. Then, $K_1 \cup K_2 \cup \{S'\} - H_1 - H_2$ forms a minimal SM-cover of B.

(8) For $D \subseteq P_1$, consider two cases: *Case* 1: $r_\mathrm{o} \notin D$. This implies that $t_\mathrm{r} \notin {}^\bullet D$ and ${}^\bullet D' = {}^\bullet D$. If $r_\mathrm{i} \notin D$, then $t_\mathrm{r} \notin D^\bullet$ and $(D')^\bullet = D^\bullet$. Hence, ${}^\bullet D' \subseteq (D')^\bullet$ provided that ${}^\bullet D \subseteq D^\bullet$. If $r_\mathrm{i} \in D$, then $t_\mathrm{r} \in D^\bullet$ and $(D')^\bullet = D^\bullet - \{t_\mathrm{r}\} \cup p_\mathrm{e}{}^\bullet$. Since ${}^\bullet D$ does not contain t_r, we have ${}^\bullet D' \subseteq (D')^\bullet$ provided that ${}^\bullet D \subseteq D^\bullet$. *Case* 2: $r_\mathrm{o} \in D$. Since ${}^\bullet D \subseteq D^\bullet$, $t_\mathrm{r} \in D^\bullet$, $\{p_i, p_\mathrm{o}\} \subseteq D'$, ${}^\bullet D' = {}^\bullet D - \{t_\mathrm{r}\} \cup {}^\bullet p_\mathrm{o}$ and $(D')^\bullet = D^\bullet - \{t_\mathrm{r}\} \cup p_\mathrm{i}{}^\bullet$. Hence, ${}^\bullet D' \subseteq (D')^\bullet$ provided that ${}_{p_\mathrm{o}}{}^\bullet \subseteq p_\mathrm{i}{}^\bullet$ in B, or equivalently, ${}^\bullet p_\mathrm{x} \subseteq p_\mathrm{e}{}^\bullet$ in B_2.

For $D \subseteq P_2$, consider two cases:

Case 1: $p_\mathrm{e} \notin D$. This implies ${}^\bullet D' = {}^\bullet D$. If $p_\mathrm{x} \notin D$, $(D')^\bullet = D^\bullet$. Hence, ${}^\bullet D' = {}^\bullet D \subseteq D^\bullet = (D')^\bullet$. If $p_\mathrm{x} \in D$, then $(D')^\bullet = D^\bullet \cup p_\mathrm{o}{}^\bullet$ and ${}^\bullet D' = {}^\bullet D \subseteq D^\bullet \subseteq (D')^\bullet$. *Case* 2: $p_\mathrm{e} \in D$. If $p_\mathrm{x} \notin D$, then $(D')^\bullet = D^\bullet$ and ${}^\bullet D' = {}^\bullet D \cup {}^\bullet p_i$. Since ${}^\bullet p_\mathrm{i} \not\subset D^\bullet = (D')^\bullet$, we have ${}^\bullet D' \not\subset (D')^\bullet$. If $p_\mathrm{x} \in D$, then $(D')^\bullet = D^\bullet \cup p_\mathrm{o}{}^\bullet$. Since ${}^\bullet D' = {}^\bullet D \cup {}^\bullet p_\mathrm{i}$, we have ${}^\bullet D' \subseteq (D')^\bullet$ provided that ${}^\bullet p_\mathrm{i} \subseteq p_\mathrm{o}{}^\bullet$ in B, or equivalently, ${}^\bullet r_\mathrm{i} \subseteq r_\mathrm{o}{}^\bullet$ in B_1.

(9) For $D \subseteq P_1$, consider two cases:

Case 1: $r_\mathrm{i} \notin D$. This implies that $t_\mathrm{r} \notin D^\bullet$ and that D gains no new output transitions, i.e., $(D')^\bullet = D^\bullet$. As for the input transitions of D, two subcases should be

distinguished: (i) $r_o \notin D$. This implies that $t_r \notin {}^\bullet D$ and that ${}^\bullet D' = {}^\bullet D$. Hence, $(D')^\bullet \subseteq {}^\bullet D'$ provided that $D^\bullet \subseteq {}^\bullet D$. (ii) $r_o \in D$. This implies that $t_r \in {}^\bullet D$ and that ${}^\bullet D' = {}^\bullet D - \{t_r\} \cup {}^\bullet p_o$. Hence, since $D^\bullet$ does not contain t_r, we have $(D')^\bullet \subseteq {}^\bullet D'$ provided that $D^\bullet \subseteq {}^\bullet D$. *Case* 2: $r_i \in D$. This implies $t_r \in D^\bullet$. If $D^\bullet \subseteq {}^\bullet D$, then $t_r \in {}^\bullet D$, $\{p_i, p_o\} \subseteq D'$, $(D')^\bullet = D^\bullet - \{t_r\} \cup p_i{}^\bullet$ and ${}^\bullet D' = {}^\bullet D - \{t_r\} \cup {}^\bullet p_o$. Hence, $(D')^\bullet \subseteq {}^\bullet D'$ provided that $p_i{}^\bullet \subseteq {}^\bullet p_o$ in B, or equivalently, $p_e{}^\bullet \subseteq {}^\bullet p_x$ in B_2.

For $D \subseteq P_2$, consider two cases:

Case 1: $p_x \notin D$. Then $(D')^\bullet = D^\bullet$. Since ${}^\bullet p_i \not\subset {}^\bullet D$, then, ${}^\bullet D \subseteq {}^\bullet D' - {}^\bullet p_i$ and $(D')^\bullet = D^\bullet \subseteq {}^\bullet D \subseteq {}^\bullet D'$.

Case 2: $p_x \in D$. Then, $(D')^\bullet = D^\bullet \cup p_o{}^\bullet$. If $p_e \in D$, then ${}^\bullet D' = {}^\bullet D \cup {}^\bullet p_i$ and $(D')^\bullet \subseteq {}^\bullet D'$ provided that $p_o{}^\bullet \subseteq {}^\bullet p_i$ in B, or equivalently, $r_o{}^\bullet \subseteq {}^\bullet r_i$ in B_1. If $p_e \notin D$, ${}^\bullet D' = {}^\bullet D$. Since $p_o{}^\bullet \notin {}^\bullet D$ and $(D')^\bullet = D^\bullet \cup p_o{}^\bullet$, $(D')^\bullet \not\subset {}^\bullet D'$.

(10) B is obtained by first deleting t_r, (r_i, t_r) and (t_r, r_o) from B_1 and then fusing r_i with p_e and r_o with p_x. This results in a total loss of three nodes and two arcs. If B_1 and B_2 are both Petri net processes, so is B. Hence, $Z(B) = |F| - |P \cup T| + 2 = (|F_1| + |F_2| - 2) - (|P_1 \cup T_1| + |P_2 \cup T_2| - 3) + 2 = |F_1| - |P_1 \cup T_1| + 2 + |F_2| - |P_2 \cup T_2| + 2 - 1 = Z(B_1) + Z(B_2) - 1$.

(11) The result is trivial.

(12) Let $M \in R(B, M_i)$. By Definition 3.2, $M(p) = M_1(p)$ if $p \in P_1$ and $M(p) = M_2(p)$ if $p \in P_2 - \{p_e, p_x\}$. By Theorem 3.1, $M_1 \in [B_1, M_r\rangle$ and $M_2 \in [B_2, M_e\rangle$. Since M_1 is bounded by k_1 and M_2 by k_2, $M(p) \leqslant \max\{k_1, k_2\}\ \forall p \in P$.

(13) $\forall M\colon M_i[B, \sigma\rangle M$, where $\sigma \in L(B, M_i)$, let $M = (m_1, m_2)$, where m_1 is the component of M over P_1 and m_2 over $P_2 - \{p_e, p_x\}$. $\forall t \in T = T_1 \cup T_2 - \{t_r\}$, consider four cases.

Case 1: $t \in T_1 - \{t_r\}$ *and* $m_2 = M_c$. This implies that $M_r[B_1, f_1(\sigma)\rangle m_1$. Since t is live in B_1, $\exists \mu : m_1[B_1, \mu t\rangle$. Let $\sigma_1 = \mu$ if μ does not contain t_r. Otherwise, let σ_1 be the result of replacing each t_r within μ with a firable e-sequence of B_2. Then, in B, we have $M_i[B, \sigma\sigma_1 t\rangle$. That is, t is live in B.

Case 2: $t \in T_1 - \{t_r\}$ and $m_2 \neq M_c$. Since B_2 is almost live and terminates properly, $\exists \sigma_2 : m_2[B_2, \sigma_2\rangle M_x$. Then, in B, we have $M_i[B, \sigma\sigma_2\rangle(m_1', M_c)$, where $m_1' = m_1 + p_o$. This becomes Case 1.

Case 3: $t \in T_2$ and $m_2 = M_c$. This implies that $M_r[B_1, f_1(\sigma)\rangle m_1$. Since t_r is live in B_1, $\exists \sigma_1 : t_r \notin \sigma_1$ and $m_1[B_1, \sigma_1\rangle m_1'[t_r\rangle$, where $m_1'(p_e) = 1$. Since t is live in B_2, $\exists \mu : M_e[B_2, \mu t\rangle$. Then, in B, we have $M_i[B, \sigma\sigma_1\mu t\rangle$. That is, t is live in B.

Case 4: $t \in T_2$ and $M_2 \neq M_c$. The proof follows from a combination of Cases 2 and 3.

(14) $\forall M\colon M_i[B, \sigma\rangle M$, where $\sigma \in \mathrm{L}(B, M_i)$, let $M = (m_1, m_2)$, where m_1 is the component of M over P_1 and m_2 over $P_2 - \{p_e, p_x\}$. Consider two cases.

Case 1: $m_2 = M_c$. Then, $m_1 = M_r[B_1, f_1(\sigma)\rangle$. Since B_1 is reversible, $\exists \sigma_1$, such that $M_1[B_1, \sigma_1\rangle M_r$. By Lemma 3.3, $\exists \sigma_1'$, such that $f_1(\sigma\sigma_1') = f_1(\sigma)\sigma_1 \in L(B_1, M_r)$. Hence, $M[B, \sigma_1'\rangle M_i$. That is, B is reversible.

Case 2: $m_2 \neq M_c$. Since B_2 terminates properly, $\exists \sigma_2\colon m_2[B_2, \sigma_2\rangle M_x$. Then, in B, we have $M[B, \sigma_2\rangle(m_1', M_c)$, where $m_1' = m_1 + p_o$. This becomes Case 1. Hence, B is reversible.

$$\begin{array}{c} \\ P-\{p_r\} \\ p_r \end{array} \begin{array}{c} T-{}^\bullet p_r-p_r^\bullet\ {}^\bullet p_r\ \ p_r^\bullet \\ \begin{pmatrix} V(P-\{p_r\},T) & & \\ 0 & I_1 & -I_2 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ P'-\{r_i,r_o\} \\ r_i \\ r_o \end{array} \begin{array}{c} T-{}^\bullet r_i-r_o^\bullet\ {}^\bullet r_i\ \ r_o^\bullet\ \ t_r \\ \begin{pmatrix} V(P-\{p_r\},T) & & & 0 \\ 0 & I_1 & 0 & -1 \\ 0 & 0 & -I_2 & 1 \end{pmatrix} \end{array}$$

Fig. .9. Incidence matrices V of B (left) and V' of B' (right). (I_1 and I_2 are identity vectors).

Proof for Theorem 4.1 (Fig. 9).

(1) According to its construction, B' obviously satisfies the definition of SM (MG, FC and AC) if B is SM (resp., MG, FC and AC).

(2) Since B is conservative, $\exists\alpha = (x_1, \ldots, x_{|P|}) \geqslant 1$ such that $\alpha V = 0$. Let $\alpha' = (x_1, \ldots, x_{|P|}, x_{|P|})$. Then, $\alpha' \geqslant 1$ and $\alpha' V' = (\alpha V, -x_{|P|} + x_{|P|}) = 0$. That is, B' is conservative. Similar argument shows that $\alpha V \leqslant 0$ leads to $\alpha' V' \leqslant 0$. Hence, B' is structurally bounded.

(3) Since B is consistent, $\exists\beta = (x_1, \ldots, x_a, y_1, \ldots, y_b, z_1, \ldots, z_c) \geqslant 1$ such that $V\beta = 0$, where $a = |T - {}^\bullet p_\mathrm{r} - p_\mathrm{r}{}^\bullet|$, $b = |{}^\bullet p_\mathrm{r}|$ and $c = |p_\mathrm{r}{}^\bullet|$. Let $y = y_1 + \cdots + y_b$ and $z = z_1 + \cdots + z_c$. Then, the bottom equation of $V\beta = 0$, i.e., $V[p_\mathrm{r}, T]\beta = 0$, leads to $y = z$. Let $\beta' = (\beta, y)$. Then, $\beta' \geqslant 1$ and $V'\beta' = 0$. The latter follows from the three horizontal components of V' : $V'[P' - \{r_\mathrm{i}, r_\mathrm{o}\}, T \cup \{t_\mathrm{r}\}]\beta' = V[P - \{p_\mathrm{r}\}, T]\beta + y \cdot 0 = 0$, $V'[r_\mathrm{i}, T \cup \{t_\mathrm{r}\}]\beta' = y_1 + y_2 + \cdots + y_a - y = 0$, and $V'[r_\mathrm{o}, T \cup \{t_\mathrm{r}\}]\beta' = -z_1 - z_2 - \cdots - z_c + y = -y + y = 0$. Hence, B' is consistent. Similar argument shows that $V\beta \geqslant 0$ leads to $V'\beta' \geqslant 0$ and that B' is repetitive.

(4)–(6) In V', after adding row r_o to row r_i, the top $|P' - \{r_\mathrm{o}\}|$ rows become $(V, 0)$ and the bottom row $(0, 0, -I_2, 1)$ is linearly independent of the other rows. The rank of this new matrix is obviously $\mathrm{rank}(V) + 1$. Hence, $\mathrm{Rank}(B') = \mathrm{Rank}(B) + 1$. In B', cluster $[p_\mathrm{r}]$ of B is replaced by cluster $[r_\mathrm{o}]$ of B' and a new cluster $[r_\mathrm{i}] = \{r_\mathrm{i}, t_\mathrm{r}\}$ is created. Hence, $|C(B')| = |C(B)| + 1$. Lastly, $\mathrm{Rank}(B') = \mathrm{Rank}(B) + 1 = |C(B)| - 1 + 1 = |C(B')| - 1$.

(7) Let K be a minimal SM-cover of B. $\forall S \in K : (p_\mathrm{r} \in S)$, create S' by replacing p_r with $\{r_\mathrm{i}, r_\mathrm{o}\} \cup \{t_\mathrm{r}\} \cup \{(r_\mathrm{i}, t_\mathrm{r})(t_\mathrm{r}, r_\mathrm{o})\}$ (with some obvious relabeling of connections) and replace S with S' in K. Since S' is a SM, the resulting K is a minimal SM-cover of B'.

(8) and (9) If $p_\mathrm{r} \notin D$, then $r_\mathrm{i} \notin D'$, $r_\mathrm{o} \notin D'$ and $t_\mathrm{r} \notin {}^\bullet D' \cup (D')^\bullet$. This implies that ${}^\bullet D' = {}^\bullet D$ and $(D')^\bullet = D^\bullet$. Hence, ${}^\bullet D' \subseteq (D')^\bullet$ provided that ${}^\bullet D \subseteq D^\bullet$ and $(D')^\bullet \subseteq {}^\bullet D'$ provided that $D^\bullet \subseteq {}^\bullet D$. If $p_\mathrm{r} \in D$, let $D' = D - \{p_\mathrm{r}\} \cup \{r_\mathrm{i}, r_\mathrm{o}\}$. Then, $(D')^\bullet = D^\bullet \cup \{t_\mathrm{r}\}$ and ${}^\bullet D' = {}^\bullet D \cup \{t_\mathrm{r}\}$. Hence, ${}^\bullet D' \subseteq (D')^\bullet$ provided that ${}^\bullet D \subseteq D^\bullet$ and $(D')^\bullet \subseteq {}^\bullet D'$ provided that $D^\bullet \subseteq {}^\bullet D$.

(10) $Z(B') = |F'| - |P' \cup T'| + 2 = |F| + 2 - (|P| + 1) - (|T| + 1) + 2 = Z(B)$.

(11) If, in B, the longest e-path containing p_r has length $\mathrm{LP}(B, p_\mathrm{r})$, then, in B', the longest e-path containing r_i and r_o has length $\mathrm{LP}(B, p_\mathrm{r}) + 1$. Since the length of those paths not containing r_i and r_o remains unchanged, $\mathrm{LP}(B') = \max\{\mathrm{LP}(B), \mathrm{LP}(B, p_\mathrm{r}) + 1\}$.

(12)–(14) Since, in B', r_i and r_o together play the same role as p_r in B, the initial marking, token distribution and firability of the transitions all remain unchanged. Hence, boundedness, liveness and reversibility of B are preserved in B'.

Proof for Theorem 4.2. By combining the corresponding properties of Theorems 3.4 and 3.5.

References

[1] W. Brauer, R. Gold, W. Vogler, A survey of behavior and equivalence preserving refinement of Petri nets, Lecture Notes in Computer Science 483 (1990) 1–46.

[2] T.Y. Cheung, Y. Lu, Detecting and resolving the interaction between telephone features Terminating Call Screening and Call Forwarding by colored Petri nets, Proc. IEEE Internat. Conf. on Systems, Man and Cybernetics, 1995, pp. 2245–2250.

[3] T.Y. Cheung, W. Zeng, Invariant-preserving transformations for the verification of place/transition systems, IEEE Trans. Systems Man Cybernet.—Part A: Systems Hum. 28 (1) (1998) 114–121.

[4] T.Y. Cheung, X. Zhu, A formal-method approach for cyclomatic complexity of distributed software, Transaction of K.C. Wong Education Foundation Supported Lectures, K.C. Wong Education Foundation, 1994, pp. 99–118.

[5] J. Desel, Place/transition Petri nets, Lecture Notes in Computer Science 1491 (1998) 122–173.

[6] J. Desel, J. Esparza, Free Choice Petri Nets, Cambridge University Press, Cambridge, 1995.

[7] J. Esparza, Reduction and synthesis of live and bounded free choice Petri nets, Inform. Comput. 114 (1) (1994) 50–87.

[8] K.X. He, M.D. Lemmon, Liveness verification of discrete event systems modeled by n-safe ordinary Petri nets, Lecture Notes in Computer Science 1825 (2000) 227–243.

[9] M.D. Jeng, F. DeCesare, A review of synthesis techniques for Petri nets with application to automated manufacturing systems, IEEE Trans. System Man Cybernet. 23 (1) (1993) 301–312.

[10] M.D. Jeng, F. DeCesare, Synthesis using resource control nets for modeling shared-resource systems, IEEE Trans. Robotics Automat. 11 (3) (1995) 317–327.

[11] W.M. Mak, Verifying property preservation for component-based software systems (a Petri-net based methodology), Ph.D. Thesis, Department of Computer Science, City University of Hong Kong, June 2001.

[12] T. Murata, Petri nets: properties, analysis and applications, Proc. IEEE 77 (4) (1989) 541–580.

[13] T. Murata, J.Y. Koh, Reduction and expansion of live and safe marked graphs, IEEE Trans. Circuits Systems 27 (1) (1980) 68–70.

[14] R.S. Pressman, Software Engineering, a Practitioner's Approach, 4th ed., McGraw-Hill Book Co., New York, 1997.

[15] M. Sliva, E. Teruel, J.M. Colom, Linear algebraic and linear programming techniques for the analysis of place/transition net systems, Lecture Notes in Computer Science 1491 (1998) 309–373.

[16] I. Suzuki, T. Murata, A method for stepwise refinement and abstraction of Petri nets, J. Comput. System Sci. 27 (1983) 51–76.

[17] R. Valette, Analysis of Petri nets by stepwise refinements, J. Comput. System Sci. 18 (1979) 35–46.