Property Analysis of Refinement of Petri Net Based Representation for Embedded Systems

Chuanliang Xia*, Zhijun Zhang and Zhong Wang

School of Computer Science and Technology, Shandong Jianzhu University, Jinan, P.R. China

Abstract: Petri net refinement is a transformation by 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 undesired properties may have been created. For systems specified in an expended Petri net, this paper proposes conditions imposed on a kind of transition subnet refinement under which timing, functionality and reachability will be preserved. Such results can be applied nicely to solve design problems in intelligent building and also enhance the property-preserving approach and characterization-based approach for system verification.

Keyword: Transformation, Petri nets, Property analysis, Equivalence, Modelling.

1. INTRODUCTION

Transformations, such as refinements, synthesis, reductions, etc., are often applied on Petri nets in order to develop a correct design specification. One of the difficult tasks, in this process, is to verify that the transformed nets possess certain desirable properties.

There exists an approach for such purposes in the literature. In this approach, the original net is assumed to be satisfying some properties and the transformation is required to preserve these properties in the transformed net. This approach is called property preservation. The advantage of property preservation is that the transformed net is automatically correct without the need of further verification.

Refinement is an important transformation. This paper investigates a special type of refinement for PRES+ and its property preservation.

In the literature, As for refinement method, Lakos [1] defined three types of refinement for coloured Petri nets: node refinement, consisting in detailing the behaviour within a place or a transition; type refinement, which addresses the refinement of data types handled; and subnet refinement which allows for the addition of a connected subnet. Völzer [2] studied the robustness of fairness notions under refinement of transitions and places in Petri nets. Choppy [3] used Coq theorem to prove the refinement relation between two Petri nets, both formally and automatically. For systems specified in pure ordinary Petri nets, Huang [4] proposed conditions imposed on several types of refinement under which 19 properties will be preserved. Xia [5, 6] proposed a kind of pp-type refinement and a kind of tt-type refinement for P/T Petri nets, and proved that these two kinds of refinements preserve some dynamic properties, such as boundedness, liveness and reversibility. Peuker [7] defined transition refinement for Algebraic Petri nets and studied how to prove that a replacement of a transition is a transition refinement. Cheung [8] proposed a refinement method for eliminating duplicate labels from a labeled net while preserving the original firing sequences (event sequences). Ahmad [9] introduced a refinement method for modeling parallel manufacturing system. Ding [10] proposed a stepwise refinement method which preserve language properties. Cabac [11] proposed a method to design a complex dynamic system at different levels of abstractions using refinements of net models. Chemaa [12] proposed an expressive object-oriented Petri net based algebra that succeeds in the complex composition of Web services, and gave a refinement operation which permits to replace certain operations of the service by more detailed ones. Köhler [13] described a special kind of Petri nets that can modify their structures via dynamic refinement of transitions. A reconfiguring process of a manufacturing system model is developed by using colored timed object-oriented Petri nets [14]. Petri nets, object-oriented methods and stepwise refinement ideas are integrated together in this model.

In order to improve verification efficiency, we proposed a set of reduction rules for PRES+ ([15]). In order to model large systems, a refinement method is needed. In this paper we will propose one kind of refinement for PRES+, and prove that this refinement method will preserve some important properties, such as reachability, timing and functionality.

The rest of the paper is organized as follows. The basic definitions of PRES+ are presented in Section 2. Then, the refinement representation method of PRES+ and its property preservation are presented in Section 3. Afterwards, a modeling example is addressed in section 4, where we will use the refinement method to model a temperature controller of Fire Protection System (FPS) in intelligent building. Conclusions are finally discussed in Section 5.
2. DEFINITIONS OF PRES+

In this section we will quickly review key definitions. A more general discussion on PRES+ can be found in [16].

Definition 2.1 [16] A PRES+ model is a five-tuple \( N = (P, T, I, O, M_0) \) where,

\[
P = \{p_1, p_2, \ldots, p_m \} \text{ is a finite non-empty set of places; } \\
T = \{t_1, t_2, \ldots, t_n \} \text{ is a finite non-empty set of transitions; } \\
I \subseteq P \times T \text{ is a finite non-empty set of input arcs which } \\
define the flow relation between places and transitions; } \\
O \subseteq T \times P \text{ is a finite non-empty set of output arcs which } \\
define the flow relation between transitions and places; } \\
M_0 \text{ is the initial marking of the net. }
\]

Definition 2.2 [16] The firing of an enabled transition \( t \in T \), for a binding \( b = (k_1, k_2, \ldots, k_n) \) changing a marking \( M \) into a new marking \( M' \). As a result of firing the transition \( t \), the following occurs:

(i) Tokens from its pre-set \( \cdot t \) are removed, that is, \( M'(p) = M(p) - \{k\} \) for all \( p, e^* t \);

(ii) One new token \( k = \langle v, r \rangle \) is added to each place of its post-set \( t^* \), that is, \( M'(p) = M(p) + \{k\} \) for all \( p \in t^* \). The token value of \( k \) is calculated by evaluating the transition function \( f \) with token values of tokens in the binding \( b \) as arguments, that is, \( v = f(v_1, v_2, \ldots, v_n) \). The token time of \( k \) is the instant at which the transition \( t \) fires, that is, \( r = tt^* \) where \( tt^* \in [tt^- tt^+] \).

(iii) The marking of places different from input and output places of \( t \) remain unchanged, that is, \( M'(p) = M(p) \) for all \( p \in P \setminus t^* \).

3. DEFINITIONS OF EQUIVALENCE

The following definitions introduce basic concepts to be used when defining the notions of property preservation for systems modeled in PRES+.

Definition 3.1 [16] A marking \( M' \) is said to be reachable from \( M \) if there exists a transition \( t \in T \) whose firing changes \( M \) into \( M' \).

Definition 3.2 [16] The reachability set \( R(N) \) of \( N \) is the set of all markings reachable from \( M_0 \); If \( M \in R(N) \) and \( M' \) is immediately reachable from \( M \), then \( M' \in R(N) \).

Definition 3.3 [16] \( N_1 \) and \( N_2 \) are said to be equivalent if the following conditions are satisfied.

(i) There exist such bijections \( f_in : inP_1 \rightarrow inP_2 \) and \( f_out : outP_1 \rightarrow outP_2 \) that define one-to-one correspondences between in(out)-ports of \( N_1 \) and \( N_2 \);

(ii) The initial markings \( M_{1,0} \) and \( M_{2,0} \) satisfy \( M_{1,0}(p) = M_{2,0}(f_{in}(p)) \neq \Phi \) \( \forall p \in inP_1 \), \( M_{1,0}(q) = M_{2,0}(f_{out}(q)) = \Phi \) \( \forall q \in outP_1 \);

(iii) For every \( M_1 \in R(N_1) \) such that \( m_1(p) = 0 \) \( \forall p \in inP_1 \), \( m_1(s) = m_2(s) \) \( \forall s \in P_1 \setminus inP_1 \setminus outP_1 \) there exists \( M_2 \in R(N_2) \) such that \( m_2(p) = 0 \) \( \forall p \in inP_2 \), \( m_2(s) = m_2(s) \) \( \forall s \in P_2 \setminus inP_2 \setminus outP_2 \), \( m_2(f_{out}(q)) = m_1(q) \) \( \forall q \in outP_1 \) and vice versa.

(iv) For every \( v_1, v_2, v_3 \in M_1(q) \), where \( q \in outP_1 \), there exists \( v_1, v_2, v_3 \in M_2(q) \) such that \( v_1 = v_2 \) and \( v_1 = v_3 \), and vice versa.

4. REFINEMENT OPERATION

In this section we propose one type subnet refinement operation for PRES+. This operation preserves equivalence. At the same time, desire properties, such as reachability, timing and functionality will be preserved.

Definition 4.1 [16] A transition \( t \in T \) is an in-transition of \( N = (P, T, I, O, M_0) \) iff \( \bigcup_{p \in t} p^* = \{t\} \). A transition \( t \in T \) is an out-transition of \( N = (P, T, I, O, M_0) \) iff \( \bigcup_{p \in t} p^* = \{t\} \).

Definition 4.2 Let \( N = (P, T, I, O, M) \) and \( N_0 = (P_0, T_0, I_0, O_0, M_0) \) be two nets. If

1. \( P_0 \subseteq P, T_0 \subseteq T \) and \( P_0 \neq \emptyset, T_0 \neq \emptyset \).
2. \( F_0 = F \setminus ((P_0 \times T_0) \cup (T_0 \times P_0)) \),
then \( N_0 \) is said to be a subnet of \( N \).

Definition 4.3 A net \( N_n = (P_n, T_n, I_n, O_n, M_{n,0}) \) is said to be a transition subnet of \( N \) if and only if,

(i) \( N_n \) is a subnet of \( N \),

(ii) \( N_n \) is connected, \( \{t_{in}, t_{out}\} \subseteq T_n \) and \( t_{in} \) is the unique in-transition of \( N_n \), \( t_{out} \) is the unique out-transition of \( N_n \),

(iii) \( inP_n \) is the set of in-ports and \( outP_n \) is the set of out-ports of \( N_n \),

(iv) \( \forall t \in T_n \setminus \{t_{in}\} \), \( t \) is disabled in \( M_{n,0} \).

Supposition 4.1 In the net \( N = (P, T, I, O, M_0) \), If \( t \in T \) is replaced by a transition subnet \( N_t = (P_t, T_t, I_t, O_t, M_{t,0}) \), then

(i) There exists a bijection \( f_{in} : t^* \rightarrow inP_t \);

(ii) There exists a bijection \( f_{out} : T^* \rightarrow outP_t \);

(iii) \( \forall p \in t^* \), \( M_t(p) = M_{t,0}(f(p)) \), \( \tau(p) = \tau(f(p)) \);

The Open Automation and Control Systems Journal, 2013, Volume 5 215
(iv) \( \forall p \in I^* \), \( M_0(p) = M_{\alpha,0}(f_{out}(p)) \) and \( \tau(p) = \tau(f_{out}(p)) \);
(v) \( \forall q \in \{ \tau_i \}, \ q \in I^* \), \( \exists v_1, r_2 \) s.t. \( M_1(q) \), \( M_2(q) \) are the in-ports of \( N_{\alpha} \);
(vi) \( \epsilon^\gamma_i \) of \( \tilde{I} \) is equal to the lower bound of the execution time of \( N_{\alpha} \);
(vii) \( \epsilon^\delta_i \) of \( \tilde{I} \) is equal to the upper bound of the execution of \( N_{\alpha} \).
We consider the net \( N \). Transition subnet \( N_\alpha = (P_\alpha, T_\alpha, I_\alpha, O_\alpha, M_{\alpha,0}) \) is a refinement of \( \tilde{I} \in T \).

**Definition 4.4** Transition subnet refinement operation \( \text{Re}_f(x, N_\alpha) \): the refinement net \( N^\prime = (P^\prime, T^\prime, I, O, M_0^\prime) \) is obtained from \( N = (P, T, I, O, M_0) \) by using \( N_\alpha = (P_\alpha, T_\alpha, I_\alpha, O_\alpha, M_{\alpha,0}) \) to replace \( \tilde{I} \) (\( \tilde{I} \in T \)), where

(i) \( P^\prime = P \cup (P_\alpha \setminus \text{in} P_\alpha \cup \text{out} P_\alpha) \);
(ii) \( T^\prime = T \cup T_\alpha \);
(iii) If \( (p, t) \in I \) or \( (p, t) \in I_\alpha \) and \( p \notin \text{in} P_\alpha \), then \( (p, t) \in I^\prime \); If \( (p, t) \in I \), then \( (p, t) \in I^\prime \);
(iv) If \( (t, p) \in O \), or \( (t, p) \in O_\alpha \) and \( p \notin \text{out} P_\alpha \), then \( (t, p) \in O^\prime \); If \( (t, p) \in O \), then \( (t, p) \in O^\prime \);
(v) For any \( p \in P \), \( M_0(p) = M_0^\prime(p) \);
\( \forall p \in P_\alpha - \text{in} P_\alpha - \text{out} P_\alpha \), \( M_0(p) = M_{1,0}(p) \).

Suppose that \( N^\prime \) is obtained from \( N \) by transition subnet refinement \( \text{Re}_f(x, N_\alpha) \).

**Definition 4.5** Let \( N_\gamma = (P_\gamma, T_\gamma, I, O, M_{\gamma,0}) \) be a subnet of \( N \), where \( P_\gamma = \tilde{I} \cup \tau^\gamma \), \( T_\gamma = \{ \tau \} \), \( I_\gamma = \{ (\tilde{p}, \tilde{i}) | p \in \tilde{I} \} \), \( O_\gamma = \{ (\tilde{p}, \tilde{p}) | p \in \tilde{I} \} \), \( M_{\gamma,0}(p) = M_{\alpha,0}(p) \) \( \forall p \in P_\gamma \).

**Definition 4.6** Let \( N_{\gamma} = (P_\gamma, T_\gamma, I, O, M_{\gamma,0}) \) be a subnet of \( N \). Where in \( N_{\gamma} \), \( P_\gamma = \tilde{I} \cup \tau^\gamma \cup \{ \{ P_\beta \setminus \text{in} P_\beta \cup \text{out} P_\beta \} \), \( T_\gamma = T_\alpha \), \( I_\gamma = \{ (t, p) | p \in \text{in} P_\beta \cup \text{out} P_\beta \} \cup I_\alpha \), \( O_\gamma = \{ (t, p) | p \in \text{in} P_\beta \cup \text{out} P_\beta \} \cup I_\alpha \), \( M_{\gamma,0}(p) = M_{\alpha,0}(p) \) \( p \in \text{in} P_\beta \cup \text{out} P_\beta \), \( M_{\gamma,0}(p) = M_{\alpha,0}(p) \) \( p \in \text{in} P_\beta \cup \text{out} P_\beta \).

**Theorem 4.1** \( N_{\gamma} \) and \( N_{\alpha} \) are equivalent.

**Proof.** By Supposition 4.1, there exists a bijection \( g_{\alpha} : \tilde{I} \rightarrow \tau^\gamma \) (where \( \tau^\gamma \) are the in-ports of \( N_{\alpha} \)); There exists a bijection \( g_{\alpha} : \tilde{I} \rightarrow \tau^\gamma \) (where \( \tau^\gamma \) are the in-ports of \( N_{\alpha} \));

Suppose \( M_{\gamma,0}(p) = M_{\alpha,0}(q) = \Phi \), \( \forall p \in \tilde{I} \) and \( M_{\alpha,0}(q) = \Phi \) for all \( p \in \tilde{I} \). By Definition 4.3 and Supposition 4.1, the initial markings \( M_{\gamma,0} \) and \( M_{\alpha,0} \) such that \( \forall p \in \tilde{I} \), \( M_{\gamma,0}(p) = M_{\alpha,0}(q) = \Phi \), \( \forall q \in \tilde{I} \), \( M_{\gamma,0}(q) = M_{\alpha,0}(q) = \Phi \).

For every \( M_{\gamma,0} \in \text{R}(N_{\gamma}) \) such that \( m_i(p) = 0 \) for all \( p \in \tilde{I} \), by Definition 4.4, there exists \( M_{\gamma,0} \in \text{R}(N_{\gamma}) \) such that \( m_i(p) = 0 \) for all \( p \in \tilde{I} \), \( m_i(q) = m_{i,0}(q) \) for all \( s \in P_{\gamma}, t_{\gamma} \in \text{out} P_{\gamma}, m_i(q) = m_{i,0}(q) \) for all \( s \in P_{\gamma}, t_{\gamma} \in \text{out} P_{\gamma}, m_i(q) = m_{i,0}(q) \).

By Definition 4.3 and Supposition 4.1, for every \( \forall p, r_1 \geq m_{i,0}(p), q \in \tilde{I} \), \( \exists v_1, r_2 \geq M_{\gamma,0}(q) \) such that \( v_1 \geq v_2 \).

Since \( \epsilon^\gamma \) of \( \tilde{I} \) is equal to the lower bound of the execution time of \( N_{\alpha} \), and \( \epsilon^\delta \) of \( \tilde{I} \) is equal to the upper bound of the execution of \( N_{\alpha} \), then \( \epsilon^\gamma \) of \( \tilde{I} \), and \( \epsilon^\delta \) of \( \tilde{I} \) are equivalent.

**Corollary 4.1** Suppose that \( N_{\gamma} \) is obtained from \( N \) by transition subnet refinement \( \text{Re}_f(x, N_\alpha) \). Then \( N_{\gamma} \) and \( N \) are equivalent.

**Proof.** Since \( N_{\gamma} = N \), \( N_{\alpha} \) and \( N \) are equivalent.

**5. APPLICATIONS**

In this section we apply the results of Section 4 to model an intelligent temperature control system of Fire Protection System (FPS) in intelligent building.

In Fig. (1), transition \( t_{\text{add}} \): system initial; \( t_{\text{K1}} \): set the upper limit temperature \( T_{\text{low}} \); \( t_{\text{K2}} \): set the lower limit temperature \( T_{\text{low}} \); \( t_{\text{med}} \): modify temperature \( T_{\text{med}} \); \( t_{\text{low}} \): modify temperature \( T_{\text{low}} \); \( t_{\text{high}} \): exchange \( T_{\text{high}} \) for \( T_{\text{low}} \); \( t_{\text{A/D}} \): A/D transformation; \( t_{\text{low}} \): transform A/D value (0-1023) into temperature value (0-100); \( t_{\text{pre}} \): to deliver out pre-alarm signal; \( t_{\text{alarm}} \): to deliver alarm signal.

When temperature \( 30^\circ \leq T < 60^\circ \), the pre-alarm signal is delivered to the alarm control panel. If temperature \( T \geq 60^\circ \), the alarm signal is delivered to the alarm control panel.

In Fig. (2), \( t_{\text{K3}}, t_{\text{K4}} \): press modify button; \( t_{\text{add}} \): add number 1; \( a > 9 \) (or \( b > 9 \)), then \( a = 0 \) (or \( b = 0 \)).
The refined net $N'$ is obtained by transition subnet refinement operation, i.e. using the transition subnet (Fig. 2) to replace transition $t_{\text{Modi1}}$ in $N$ (Fig. 1). By Corollary 4.1, $N'$ and $N$ are equivalent.

So, in $N'$ the properties, such as reachability, timing and functionality of $N$ are preserved. Note that other transitions, such as $t_{\text{modi2}}$ and $t_{\text{tran}}$, can also be refined by transition subnet refinement operation.

6. CONCLUSIONS

In this paper we investigate property preservations of transition subnet refinement operation. Based on PRES+ model this paper proposes conditions under which reachability, timing and functionality will be preserved. These results can be applied nicely to solve design problems in intelligent building, manufacturing engineering and software engineering, and also release the designer's burden for having to provide different methods for individual properties. Further research is needed to investigate more general subnet refinement operations for PRES+ and their applications.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the National Natural Science Foundation of China under Grant No. 60573012 and 60721061, the National Grand Fundamental Research 973 Program of China under Grant No. 2002cb312200, and Natural Science Foundation of Shandong Province under Grant No. ZR2012GQ010.
REFERENCES


Received: August 13, 2013  Revised: August 28, 2013  Accepted: August 28, 2013

© Xia et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.