Testing Of Functions Of Complex Systems Based On Synchronous Composition Nets[1]

Changjun Jiang

Department of Computer Science and Engineering

Tongji University

Shanghai 200092

THE PEOPLE' S REPUBLIC OF CHINA

Abstract: In this paper, two essential testing methods for firing sequence of Petri nets are presented. The first testing method is to test whether the firing sequences in two subsystems can be shuffled into their composition system. The second method is for testing whether a given sequence belongs to a composition system, by testing whether the sequence belongs to the subsystems of the composition system. An example has been proposed using our new methodology, to demonstrate the advantages of this methodology.

Keywords: Functions testing, synchronous composition nets, firing sequence

Changjun Jiang received his BE and ME from University of Shandong Science and Technology in 1986 and 1991, respectively. He received his Ph.D degree from Institute of Automation Chinese Academy of Sciences. He is currently Professor at the Department of Computer Science and Engineering, Shanghai Tongji University. His areas of interest in research include Petri net, concurrent system, model checking and fuzzy reasoning.

1. Introduction

A composition via a set of places consists simply of merging distinguished places of two nets. This kind of operation has already been used in [1]. [2] and [3] discussed composition of two nets via the set of share transitions of the two nets. [4] and [5] described composition of two nets via their common subnet of the two nets under their structural meaning. In the method of dependency analysis, [6] synthesised a large Petri Net by combining smaller ones. Whether the composition preserves the properties depends on the dependency relations among the transitions. [7] presented the dynamic analysis method based on the reduction of the reachability graph proposed. The idea of all these techniques above is to perform a global system analysis from the analysis of its subsystems.

The set of all firing sequences of a Petri net is an important tool for describing the dynamic behaviour of concurrent systems. For example, the temporal Petri nets model is the integration of the general Petri net and the temporal logic [8, 9]. The firing sequences of such kind of Petri nets are analysed under temporal logic conditions. An analysis method with the help of ω-regular expressions and Buchi-automata is presented in [9], based on the reachability graph of Petri nets. Paper [10] used temporal Petri nets as a modelling tool, and studied the specification and verification of multi-axis high-speed machines. Paper [11] presented the method of mapping Petri nets with inhibitor arcs onto basic LOTOS behaviour expressions.

By comparison with [7, 8, 9], our analysis method is based on T-invariants and our testing methods are based on the synchronous composition of Petri nets and the reachability graph of subnets. On the other hand, their methods are based on the reduction reachability graph [7] or the reachability graph of Petri net and automata method [8, 9]. Compared with [6], our study aims at the relativity of subnets under synchronous composition operation for Petri nets. The aim from [6] is the liveness of systems under a kind of knitting operation for Petri nets.

This paper is organized as follows. Section 2 presents the basic concepts and terminology for Petri nets. The behaviour properties of synchronous composition nets are discussed in Section 3. Two testing algorithms are given in

1 *Supported by National Natural Science Foundation of P.R.China Excellent Ph.D Paper Author Foundation of P.R.China Dawn Plan Foundation of Shanghai and Excellent Young Scientist Foundation of Shandong Province.*

Section 4, respectively. In order to evaluate our methodology, an example is discussed in Section 5. Section 6 is meant for the conclusion.

2. Basic Concepts and Terminology

A triple $N = (P, T; F)$ is called a net iff

(1) $P \cap T = \varnothing, P \cup T \neq \varnothing$;

(2) $F \subseteq (P \times T) \cup (T \times P)$; and

(3) $dom(F) \cup con(F) = P \cup T$.

For $\forall x \in P \cup T$,

${}^{\bullet}x = \{y | (y \in P \times T) \wedge ((y, x) \in F)\}$ and

$x^{\bullet} = \{y | (y \in P \times T) \wedge ((x, y) \in F)\}$ are called the pre-set and the post-set of x, respectively.

$\Sigma = (N, M_0) = (P, T; F, M_0)$ is called a Petri net iff

(1) $N = (P, T; F)$ is a net;

(2) $M_0 : P \to Z$ (set of non-negative integers) is called an initial marking of N (or Σ); and

(3) the following firing rules apply:

(3.1) $t \in T$, M is a marking of Σ, t is said to be M-enabled (denoted as $M[t >$) iff $\forall p \in {}^{\bullet}t \cap P$: $M(p) > 0$; and

(3.2) t can be fired from M if t is M-enabled (denoted as $M[t >$). Firing t from M results in a new marking M' (denoted as $M[t > M'$), for $\forall p \in P$, we have

$$M'(p) = \begin{cases} M(p) + 1, & if\ p \in t^{\bullet} - {}^{\bullet}t; \\ M(p) - 1, & if\ p \in {}^{\bullet}t - t^{\bullet}; \\ M(p), & otherwise. \end{cases}$$

If $\exists t_1, t_2, \ldots, t_i \in T$ and markings $M_0, M_1, \ldots, M_i$, such that

$$M_0[t_1 > M_1[t_2 > M_2 \ \ldots \ M_{i-1}[t_i > M_i,$$

then marking M_i is said to be reachable from M_0, denoted $M_i \in R(M_0)$ or $M_0[\sigma > M_i$, where $\sigma = t_1 t_2 \ldots t_i$ is called a transition sequence from M_0 to M_i. And $R(M_0)$ is called the set of all reachable markings from M_0.

Let $L(\Sigma) = \{\sigma | (\sigma \in T^*) \wedge (M_0[\sigma >)\}$, then $L(\Sigma)$ is called the language of Σ.

The synchronous composition is a well-known operation in Petri nets research. The properties and the analysis method of synchronous composition net such as reachability, deadlock, as well as how to construct and to reduce state space of composition net from state spaces sub-nets have been studied [2, 3]. However, few researches have focused on their sequences testing aspects, which we will be discussed in this paper.

There are two Petri nets $\Sigma_i = (P_i, T_i; F_i, M_{0i}), i = 1,2,$. Suppose that these two nets are live and bounded. Now we make a new net Σ by connecting those common transitions. The goals of our research are

(P1) How to test $\sigma_{1j} \otimes \sigma_{2j} \subseteq L(\Sigma)?\ j = 1,2,\ldots,k$, assume $\sigma_{ji} \in L(\Sigma_i), i = 1,2$;

(P2) How to test $\sigma_j \in L(\Sigma)?\ j = 1,2,\ldots,k$.?

3. Behaviour Properties of Synchronous Composition Nets

A number of formal definitions are shown as follows. They are essential to our further discussion.

Definition 1

Let $\Sigma_i = (P_i, T_i; F_i, M_{0i}), i = 1,2,$ be two Petri nets, $P_1 \cap P_2 = \emptyset$, and $T_1 \cap T_2 \neq \emptyset$. Set $\Sigma = (P, T; F, M_0)$, such that

(1) $P = P_1 \cup P_2$;

(2) $T = T_1 \cup T_2$;

(3) $F = F_1 \cup F_2$; and

(4) $M_0(p) = M_{0i}(p), \text{ if } p \in P_i, i = 1,2$

then Σ is called a synchronous composition net of Σ_1 and Σ_2, denoted $\Sigma = \Sigma_1 \oplus \Sigma_2$.

Definition 2

Let X be a finite alphabet, $Y \subseteq X$. (1) Set $\Gamma_{X \to Y} : X^* \to Y^*$, such that $\forall \sigma \in X^*$, and $\Gamma_{X \to Y}(\sigma)$ is the remnant sub-string after deleting each element in $X - Y$ from σ. $\Gamma_{X \to Y}$ is called a projection mapping from X to Y. (2) Set $\Gamma^{-1}_{Y \to X} : Y^* \to X^*$, such that $\forall \sigma' \in Y^*$, and

$$\Gamma^{-1}_{Y \to X}(\sigma') = \{\sigma \mid (\sigma \in X^*) \wedge (\Gamma_{X \to Y}(\sigma) = \sigma')\}$$

. $\Gamma^{-1}_{Y \to X}$ is called an extension mapping from Y to X, where " $*$ " is a closed operation of the language.

Definition 3

Let X be a finite alphabet, $Y \subseteq X$, L_X and L_Y be the languages on X and Y, respectively. Set $\Gamma_{X \to Y}(L_X) = \{\Gamma_{X \to Y}(\sigma) \in Y^* \mid \forall \sigma \in L_X\}$ and $\Gamma^{-1}_{Y \to X}(L_Y) = \bigcup_{\forall \sigma' \in L_Y} \Gamma^{-1}_{Y \to X}(\sigma')$,

then $\Gamma_{X \to Y}(L_X)$ and $\Gamma^{-1}_{Y \to X}(L_Y)$ are called the projection language of L_X from X to Y, and the extension language of L_Y from X to Y, respectively. Then $\Gamma_{X \to Y}(L_X)$ and $\Gamma^{-1}_{Y \to X}(L_Y)$ are called projection language of L_X from X to Y, and extension language of L_Y from Y to X, respectively.

For example :

If

$X = \{a,b,c\}, L(X) = \{abba, babbc, bbcaa\}, X_s = \{a,c\}$, then

$\Gamma_{X \to X_s}(L(X)) = \{aa, ac, caa\}$.

Theorem 1

Let $\Sigma_i = (P_i, T_i; F_i, M_{0i}), i = 1,2,$ be two Petri nets, and $\Sigma = \Sigma_1 \oplus \Sigma_2$. Then $L(\Sigma) = \Gamma^{-1}_{T_1 \to T}(L(\Sigma_1)) \cap \Gamma^{-1}_{T_2 \to T}(L(\Sigma_2))$.

Proof

For convenience suppose the empty letter $\varepsilon \notin T$, and

(i) $M[\varepsilon > M' \Leftrightarrow M = M'$,

(ii) $\forall t \in T, t \bullet \varepsilon = t$.

$\sigma \in L(PN)$ iff

$\exists M_0, M_1, .., M_k \in R(M_0)$, such that

$M_0[\sigma(1) > M_1[\sigma(2) > M_2 \ldots M_{k-1}[\sigma(k) > M_k$
where

$\sigma = \sigma(1) \bullet \sigma(2) \bullet \ldots \bullet \sigma(k)$.

Let $M_i = (M_{i1}^T, M_{i2}^T)^T, \ i = 0, 1, 2, \ldots, k$. iff

$$\left[\left((M_{i1}^T, M_{i2}^T)^T \geq \left((C_1^- \sigma(i+1))^T, 0^T\right)^T\right) \wedge \left(\begin{array}{l}(M_{i+1,1}^T, M_{i+1,2}^T)^T = \\ (M_{i1}^T, M_{i2}^T)^T + \left((C_1 \underline{\sigma(i+1)})^T, 0^T\right)^T\end{array}\right)\right]$$

$$\vee\left[\begin{pmatrix}(M_{i1}^T, M_{i2}^T)^T \geq \\ (0^T, (C_2^T \underline{\sigma(i+1)})^T)^T\end{pmatrix} \wedge\right.$$

$$\left.\begin{pmatrix}(M_{i+1,1}^T, M_{i+1,2}^T)^T = \\ (M_{i1}^T, M_{i2}^T)^T + ((C_2 \underline{\sigma(i+1)})^T, 0^T)^T\end{pmatrix}\right]$$

$$\vee\left[\begin{pmatrix}(M_{i1}^T, M_{i2}^T)^T \geq \\ ((C_1^- \underline{\sigma(i+1)})^T, (C_2^- \underline{\sigma(i+1)})^T)^T\end{pmatrix} \wedge\right.$$

$$\left.\begin{pmatrix}(M_{i+1,1}^T, M_{i+1,2}^T)^T = \\ (M_{i1}^T, M_{i2}^T)^T + \quad ((C_1^- \underline{\sigma(i+1)})^T, (C_2^- \underline{\sigma(i+1)})^T)^T\end{pmatrix}\right]$$

where $C_j = C_j^+ - C_j^-$ is the incidence matrix of Σ_j, $\underline{\sigma(i+1)}$ indicates $|T_j|$-vectors $(00\ldots1\ldots0)^T$, $i = 1, 2, \ldots, k;\ \ j = 1, 2$.

iff

$$\left[\begin{matrix}(M_{i1}[\sigma_1(i+1) > M_{i+1,1}) \wedge (M_{i2} = \\ M_{i+1,2}) \wedge (\sigma_1(i+1) =\end{matrix}\right.$$

$$\left.\sigma(i+1)) \wedge (\sigma_2(i+1) = \varepsilon)\right]$$

$$\vee\left[\begin{matrix}(M_{i2}[\sigma_2(i+1) > M_{i+1,2}) \wedge (M_{i1} = \\ M_{i+1,1}) \wedge (\sigma_2(i+1) =\end{matrix}\right.$$

$$\left.\sigma(i+1)) \wedge (\sigma_1(i+1) = \varepsilon)\right]$$

$$\vee\left[\begin{matrix}(M_{i1}[\sigma_1(i+1) > M_{i+1,1}) \wedge \\ (M_{i2}[\sigma_2(i+1) > M_{i+1,2}) \wedge\end{matrix}\right.$$

$$\left.(\sigma_1(i+1) = \sigma_2(i+1) = \sigma(i+1))\right]$$

iff

$$(M_0[\sigma_1 > M_{k,1}) \wedge (M_{i2}[\sigma_2 > M_{k,2}) \wedge$$
$$(\sigma_j(1) \bullet \sigma_j(2)$$
$$\bullet \ldots \bullet \sigma_j(k) = \sigma_j) \wedge (j = 1,2).$$

iff

$$(\sigma_j \in L(\Sigma_j)) \wedge (\sigma_j = \Gamma_{T \to T_j}(\sigma)) \wedge (j = 1, 2).$$

iff

$$(\sigma \in (\Gamma_{T_j \to T}^{-1}(L(\Sigma_j)) \wedge (j = 1, 2).$$

iff

$$\sigma \in \Gamma_{T_1 \to T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \to T}^{-1}(L(\Sigma_2)).$$

Hence

$$L(\Sigma) = \Gamma_{T \to T_1}^{-1}(L(\Sigma_1)) \cap \Gamma_{T \to T_2}^{-1}(L(\Sigma_2)) \quad \square$$

Lemma 1

Let $\Sigma_i = (P_i, T_i; F_i, M_{0i})$, $i = 1,2$, be two Petri nets, $\Sigma = \Sigma_1 \oplus \Sigma_2$, and $\Delta = T_1 \cap T_2$.

Then

$$\Gamma_{T \to \Delta}(L(\Sigma)) = \Gamma_{T_1 \to \Delta}(L(\Sigma_1)) \cap \Gamma_{T_2 \to \Delta}(L(\Sigma_2)).$$

Proof

From theorem 1,

$$L(\Sigma) = \Gamma_{T_1 \to T}^{-1}(L(\Sigma_1)) \cap \Gamma_{T_2 \to T}^{-1}(L(\Sigma_2)).$$

Thus

$$\Gamma_{T \to \Delta}(L(\Sigma)) =$$
$$\Gamma_{T \to \Delta}(\Gamma_{T_1 \to T}^{-1}(L(\Sigma_1))) \cap \Gamma_{T \to \Delta}(\Gamma_{T_2 \to T}^{-1}(L(\Sigma_2))) =$$
$$= \Gamma_{T_1 \to \Delta}(L(\Sigma_1)) \cap \Gamma_{T_2 \to \Delta}(L(\Sigma_2)).$$

Hence Lemma 1 is proven. □

4. Testing Algorithms of Firing Sequences

In order to verify system functions,. sequences presenting system functions are tested to see whether these sequences belong to the target system [8, 9]. Testing firing sequences of complex systems is often a difficult task. Decomposing complex systems and deconcentrating test firing sequences prove to be effective. Our idea for such testing is based on Theorems 2 and 3 shown below.

Theorem 2

Let $\Sigma_i = (P_i, T_i; F_i, M_{0i}), i = 1,2$ be two live Petri nets, $\Sigma = \Sigma_1 \oplus \Sigma_2$, and $\Delta = T_1 \cap T_2$. If $\sigma_i \in L(\Sigma_i), i = 1,2$, then $\Gamma_{T_1 \to \Delta}(\sigma_1) = \Gamma_{T_2 \to \Delta}(\sigma_2)$, iff $\sigma_1 \otimes \sigma_2 \subseteq L(\Sigma)$.

Proof

Since $\sigma_i \in L(\Sigma_i), i = 1,2$, then

$\Gamma_{T_1 \to \Delta}(\sigma_1) = \Gamma_{T_2 \to \Delta}(\sigma_2)$,

iff $\forall \sigma \in \sigma_1 \otimes \sigma_2$:
$\Gamma_{T \to T_i}(\sigma) \in L(\Sigma_i), i = 1,2$.

iff $\sigma_1 \otimes \sigma_2 \subseteq \Gamma^{-1}_{T_i \to T}(L(\Sigma_i)), i = 1,2$.

iff

$\sigma_1 \otimes \sigma_2 \subseteq \Gamma^{-1}_{T_1 \to T}(L(\Sigma_1)) \cap \Gamma^{-1}_{T_2 \to T}(L(\Sigma_2))$.

From Theorem 1, we have $\sigma_1 \otimes \sigma_2 \subseteq L(\Sigma)$.

Theorem 3

Let $\Sigma_i = (P_i, T_i; F_i, M_{0i}), i = 1,2$ be two Petri nets. $\Sigma = \Sigma_1 \oplus \Sigma_2$, and $\Delta = T_1 \cap T_2$.

Then $\sigma_i = \Gamma_{T \to T_i}(\sigma) \in L(\Sigma_i), i = 1,2$, iff $\sigma \in L(\Sigma)$.

Proof

It is easy to see if this consequence is true by applying Lemma 1, Theorem 1 and Theorem 2.

Based on Theorem 2 and Theorem 3, the testing algorithms for a given group of firing sequences are described as follows.

Algorithm 1. *Shufting_Test*

Input $\sigma_{ij} \in L(\Sigma_i), i = 1,2; j = 1,2,\ldots,q$

Output $b(j), j = 1,2,\ldots,q$ / if $\sigma_{1j} \otimes \sigma_{2j} \subseteq L(\Sigma_1 \oplus \Sigma_2)$ then $b(j) = 1$, otherwise $b(j) = 0$. /

(1) begin

(2) for $j = 1$ to q do

(3) if $\Gamma_{T_1 \to \Delta}(\sigma_{1j}) = \Gamma_{T_2 \to \Delta}(\sigma_{2j})$ then

(4) $b(j) \leftarrow 1$

(5) else $b(j) \leftarrow 0$

(6) endif

(7) endfor

(8) endbegin.

Algorithm 2. *Belonging_Test*

Input $\sigma_j, j = 1,2,\ldots,q; \Sigma_i, i = 1,2$

Output $b(j), j = 1,2,\ldots,q$ / if $\sigma_j \in L(\Sigma_1 \oplus \Sigma_2)$ then $b(j) = 1$, otherwise $b(j) = 0$ /

(1) begin

(2) Generate reachability graphs $RMG(\Sigma_i)$ of $\Sigma_i, i = 1,2$; based on the algorithm of well-known reachability graphs for bounded Petri nets [12];

(3) for $j = 1$ to q do

(4) if $\Gamma_{T \to T_i}(\sigma) \in L(\Sigma_i), i = 1,2$ then

(5) $b(j) \leftarrow 1$

(6) else $b(j) \leftarrow 0$

(7) endif

(8) endfor

(9)endbegin.

It is clear that the *Shufting_Test* algorithm needs only a few projections and comparisons, while the *Belonging_Test* algorithm needs only a few projections and tests for subsystems. Therefore, the complexity of testing large systems has been reduced.

5. Example

In order to evaluate our analysis methodology described above, two well-known examples

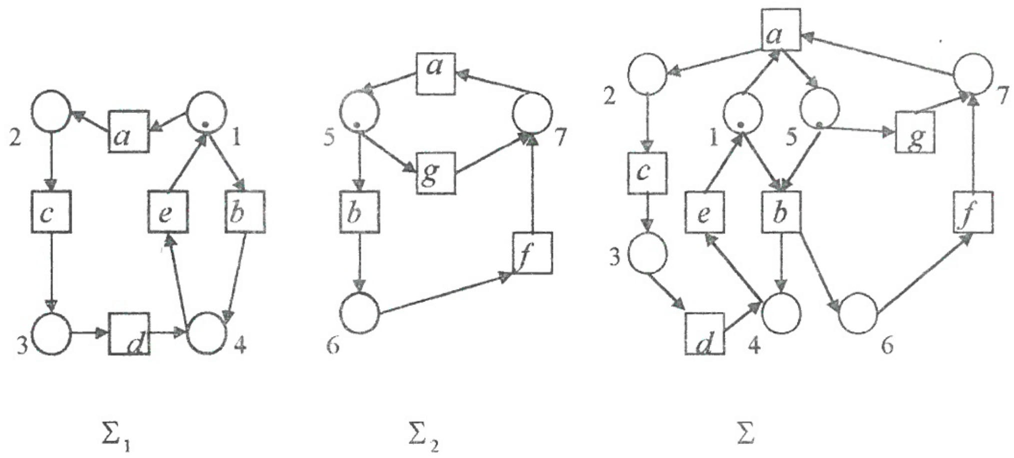

Figure 1. Synchronous Composition of Petri Nets

have been selected to demonstrate the application of our methodology.

Example. Two Petri nets Σ_i are shown in Figure 1, and their synchronous composition net is also shown in Figure 1. Two new problems are considered as follows.

respectively. It is necessary to test whether $\sigma_{1j} \otimes \sigma_{2j} \subseteq L(\Sigma)$ are true, where

$\Sigma = \Sigma_1 \otimes \Sigma_2$ (see Figure 1). By applying algorithm *Shufting_Test* described previously, the result (when $q = 10$) is given in Table 1.

Table 1. The Testing Result of the Algorithm *Shufting_Test* for 10 Sequences

σ_{1j}	σ_{2j}	$\Gamma_{T_1\to\Delta}(\sigma_{j1})$	$\Gamma_{T_2\to\Delta}(\sigma_{2j})$	$\Gamma_{T_1\to\Delta}(\sigma_{j1}) = \Gamma_{T_2\to\Delta}(\sigma_{2j})$?	$\sigma_{1j} \otimes \sigma_{2j} \subseteq L(\Sigma)$?
aceba	gabfa	aba	aba	$=$	$\subseteq$
bebeac	bfab	bba	bab	$\neq$	no $\subseteq$
acdebe	bfaga	ab	baa	$\neq$	no $\subseteq$
beacde	bfag	ba	ba	$=$	$\subseteq$
beacde	bfaga	ba	baa	$\neq$	no $\subseteq$
acdeb	gabf	ab	ab	$=$	$\subseteq$
acdeb	gebfag	ab	aba	$\neq$	no $\subseteq$
beacdea	bfag	baa	ba	$\neq$	no $\subseteq$
beac	bfag	ba	ba	$=$	$\subseteq$
bebea	bfabf	bba	bab	$\neq$	no $\subseteq$

Problem 1

The Petri nets Σ_i in Figure 1 have a group of sequences σ_{ij}, $j = 1,2,\ldots,q$; $i = 1,2$,

Problem 2

The Petri nets Σ in Figure 1 has a group of sequences σ_j, $j = 1, 2, \ldots, q$. It is necessary

Table2. The Testing Result of the Algorithm *Belonging_Test* for 10 Sequences

σ_j	$\Gamma_{T\to T_1}(\sigma_j)$	$\Gamma_{T\to T_2}(\sigma_j)$	$\Gamma_{T\to T_1}(\sigma_j)\in L(\Sigma_1)$?	$\Gamma_{T\to T_2}(\sigma_j)\in L(\Sigma_2)$?	$\sigma_j \in L(\Sigma)$?
abefabes	abeabe	abfabf	$\notin$	$\in$	$\notin$
befacdea	beacdea	bfaa	$\in$	$\notin$	$\notin$
gacgdeacde	acdeacde	gaga	$\in$	$\in$	$\in$
gacbed	acbed	gab	$\notin$	$\in$	$\notin$
bfdecabc	bdecabc	bfab	$\notin$	$\in$	$\notin$
bfeacdea	beacdea	bfaa	$\in$	$\notin$	$\notin$
fgcdeac	cdeac	fga	$\notin$	$\notin$	$\notin$
befagcdeac	beacdeac	bfaga	$\in$	$\in$	$\in$
gacgde	acde	gag	$\in$	$\in$	$\in$
gacdega	acdea	gaga	$\in$	$\in$	$\in$

to test whether $\sigma_j \in L(\Sigma), j = 1, 2, \ldots, q$ are true, where $\Sigma = \Sigma_1 \otimes \Sigma_2$ (see Figure 1). By applying algorithm *Belonging_Test* described previously, the result (when $q = 10$) is given in Table 2.

6. Conclusions

Two new sequences testing methods are presented in this paper. With the help of the behaviour characteristics, we need to test the sequences between the subsystems, rather than between a composition system and its subsystems. Such recursive analysis methods can reduce the complexity of system analysis.

At present, the authors are applying this methodology to the verification of parallel programming. We are also interested in further extending our research, such as verifying system functions of temporal Petri nets.

REFERENCES

1. SOUISSI, Y., **On Liveness Preservation By Compositional of Nets Via a Set of Places**, Lecture Notes on Computer Science, Vol. 524, SPRINGER-VERLAG, 1991, pp. 277-295.
2. NOTOMI, M. and MURATA, T., **Hierarchical Reachability Graph of Bounded Petri Nets for Concurrent Software Analysis**, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 20, No. 5, 1994, pp. 325-336.
3. VALMARI, A., **Compositional State Space Generation**, Lecture Notes on Computer Science, Vol. 674, SPRINGER-VERLAG, 1993, pp. 427-457.
4. JIANG, C. J. and WU, Z. H., **Nets Operations**, JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY, Vol. 7, No. 4, 1992, pp. 333-344.
5. JIANG, C. J., **Nets Operations (II)**, JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY, Vol. 10, No. 6, 1995, pp. 509-517.

6. CHEN, Y., TSAI, W.T. and CHAO, D., **Dependency Analysis - A Petri Net Based Technique for Synthesizing Large Concurrent Systems**, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Vol. 4, No. 4, 1993, pp. 414-426.

7. WILLSON, R. G. and KROGH, B.H., **Petri Net Tools for the Specification and Analysis of Discrete Controllers**, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 16, No. 1, 1990, pp. 39-50.

8. SUZUKI, I., **Formal Analysis of the Alternating Bit Protocol by Temporal Petri Nets**, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 16, No. 11, 1990, pp. 1273-1281.

9. SUZUKI, I. and LU, H., **Temporal Petri Nets and Their Application to Modelling and Analysis of A Handshake Daisy Chain Arbiter**, IEEE TRANSACTIONS ON COMPUTING, Vol. 38, No. 5, 1989, pp. 696-704.

10. JIANG, C. J. ET AL, **Real-time Synchronisation of Multiaxis High-speed Machines, SFC Specification to Petri Net Verification**, IEE Proceedings on Systems Engineering For Automation, Vol. 143, No. 2, 1996, pp. 164-170.

11. SISTO, R. and VALENZANO, A., **Mapping Petri Nets With Inhibitor Arcs Onto Basic LOTOS Behavior Expressions**, IEEE TRANSACTIONS ON COMPUTING, Vol. 44, No. 12, 1995, pp. 1361-1370.

12. MURATA, T., **Petri Nets: Properties, Analysis and Applications**, Proceedings of IEEE, Vol. 77, 1989, pp. 541-580.