Research on transition priorities in group based on CPN

Hong Wang1,2,3*, Tao Zhang2

1Academy of OPTO-Electronics, Chinese Academy of Sciences, Deng Zhaung Rd. 9, Distinct Haidian, Beijing, China
2Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Deng Zhaung Rd. 9, Distinct Haidian, Beijing, China
3University of Chinese Academy of Sciences, Yuquan Rd. 19, Shijingshan District, Beijing, China

Received 1 March 2014, www.cmnl.cn

Abstract

Transition priorities might be a useful mechanism when modelling using Petri nets. Accordingly, the newest CPN Tools, widely used for modelling and simulating the Coloured Petri net, implements transition priorities. Whereas, the algorithms compute enabling for all transitions in a highest-priority-first order. In the real system, it is usually that there are priorities relationships not for all transitions but only some ones. Based on the above analysis, this paper put all the transitions, having priority relations, into one group and advances relative theoretical definitions of transition priorities in group, such as absolute preset of transition, key place set, key colour set, etc. Furthermore, it proposes new algorithms when the systems have different key place set and key colour set, and construct the model of the interrupt priorities to solve the problem of software model checking for interrupt system.

Keywords: Petri net, CPN, transitions, transition priorities in group, counter place

1 Introduction

In the process of development of Petri network, in order to better describe realistic system, researchers advanced various high-level Petri net models, such as: time Petri nets [1,2], Stochastic Petri nets [3,4], Colored Petri nets [5,6], Priorities Petri nets [7], Hybrid Petri nets [8], and so on. These extended concepts enhanced Petri net description of capabilities, furthermore, modelling various Petri net had been widely applied in the synchronous system, asynchronous system, simulation [9, 10] and analysis [11, 12] process. However, the actual applications need a better tool to simulate and verity generated models. CPN Tools widen the usage of Petri nets in practical work [13-15].

CPN Tools version 4.0 [16], supporting transition priorities, could specify priority P_HIGH, P_NORMAL or P_LOW [17] for transitions, and could also directly give an integer. Nevertheless, this kind of priorities is static and absolute priority for all transitions, precedence relations, in the actual system, exist among not all transitions but some of them.

Reference [18] proposed to increase one Anti-Place to address the transition priorities, shown in FIGURE 1. The Anti-Place is effective, only if the place had tokens in initial state, however, most systems could not meet the mentioned requirements. Reference [19] used the concept of generalized complement position (Place), changing Petri net system with dynamic priority into the Petri net system with non-priority, but the algorithm was too complex even for constructing a simple system.

![FIGURE 1 Anti-place for transitions with priorities](Image)

The remainder of this paper is structured as follows: in the next section, it presented background material and in Section III it provides the definitions of transition priorities in group, key place set and key colour set, etc., and In Section IV, it presents algorithms for increasing one “counter place” to calculate the change of tokens in key place set. Furthermore, the next section, the article solve the practical work interrupt relational model by using the above definitions and algorithms. In Section VI, it concludes and provides directions for future works.

2 Petry net and coloured Petri net

2.1. PETRI NET (PLACE/TRANSITION NET OR P/T NET).

A Petri net is one of several mathematical modelling languages for the description of concurrent systems. A Petri net is a directed bipartite graph, in which the nodes represent transitions and places. The directed arcs described which places are pre- and/or post- conditions, for which transitions (signified by arrows).

*Corresponding author e-mail: neiep_wh@163.com
A Petri net is a 5-triple \( PN = (P, T, F, W, M_0) \) [20], where:

(i) \( P = \{ p_1, p_2, p_3, \ldots, p_n \} \) is a finite set of places.
(ii) \( T = \{ t_1, t_2, t_3, \ldots, t_s \} \) is a finite set of transitions.
(iii) \( F \subseteq (P \times T) \cup (T \times P) \) is a set of arcs (flow relation).
(iv) \( W : F \rightarrow \{ 1, 2, 3, \ldots \} \) is a weight initial marking.
(v) \( M_0 : P \rightarrow \{ 1, 2, 3, \ldots \} \) is an initial marking.

\( P \cap T = \emptyset \) and \( P \cup T \neq \emptyset \).

The **preset** of a transition or place \( x \) is the set of its input places or transition: \( x^* = \{ y \mid (y, x) \in E \} \); its **postset** is the set of its output places or transition: \( x^* = \{ y \mid (x, y) \in E \} \).

The **preset** of a set \( X_1 \subseteq P \), where \( X_1 \subseteq \{ x \mid x \in X_1 \} \) is also defined as \( x^* = \bigcup_{x \in X_1} x^* \).

However, basic Petri nets have limitations. One of the limitations is homogenous tokens: the tokens inside place represent resources; then in Petri nets, all these resources are of the same type and cannot be differentiated. Another limitation of Petri net is that it is not possible to impose additional logical functions (firing conditions) for a transition.

### 2.2 COLOURED PETRI NETS (CP-NETS OR CPN)

Coloured Petri nets are one of popular extensions of basic Petri net. In a standard Petri net, tokens are indistinguishable. Nevertheless, in a Coloured Petri Net, each token has a value, which could be described with colour set.

A CP-net is a 9-tuple \( CPN = (\Sigma, P, T, A, N, C, G, E, I) \) [21], where:

(i) \( \Sigma \) is a finite set of non-empty types, also called colour sets;
(ii) \( P \) is a finite set of places;
(iii) \( T \) is a finite set of transitions;
(iv) \( A \) is a finite set of arcs such that:

\[
P \cap T = P \cap N = T \cap A = \emptyset ;
\]

(v) \( N \) is a node function. It is defined from \( A \) into \( P \times T \cup T \times P \);
(vi) \( C \) is a colour function. It is defined from \( P \) into \( E \);
(vii) \( G \) is a guard function. It is defined from \( T \) into expressions such that:

\[
\forall t \in T : T[Type(G(t)) = Boolean \land Type(Var(G(t))) \subseteq \Sigma] ;
\]

(viii) \( E \) is an arc expression function. It is defined from \( A \) into expressions such that:

\[
\forall a \in A : T[Type(E(a)) = C(p) \land Type(Var(E(a))) \subseteq \Sigma] ,
\]

where \( p \) is the place of \( N(a) \);

(ix) \( I \) is an initialisation function. It is defined from \( P \) into closed expressions.

Such that:

\[
\forall p \in P : T[Type(I(p)) = C(p)_M] .
\]

In popular tools for Coloured Petri nets, such as CPN Tools, the values of tokens are typed, and could be tested (using guard functions) and manipulated with a functional programming language.

### 3 Definitions of transitions priorities in group

In order to solve transitions priorities in group, the relative definitions listed as follows.

#### 3.1. \( P_r(t) \) (TRANSITION PRIORITIES IN GROUP)

When one transition \( t \) is enabled, there is another transition \( t_i \) (\( t_i \neq t \)) could also be enabled, but they have different priorities. A group set is composed of all transitions, which could be enabled at the same time and have different priorities. The group set \( S = \{ t, t_i \} \), the transition property of \( t \) in group set \( S \) is called \( P_r(t) \). Similarly, the transition property of \( t_i \) in group set \( S \) is \( P_r(t_i) \).

#### 3.2 \( T_r \) (GROUP OF TRANSITIONS WITH PRIORITIES)

When multiple transitions could be enabled at the same time, there is relationship of priorities among transitions in the group. The group is an ordered collection of the sort, according to the priorities from high to low. Defined as follows:

A CP-net with Transitions priorities is a 10-tuple \( CPN' = (\Sigma, P, T, A, N, C, G, E, I, S) \), where:

\( S \) is a finite set of group of enabled transitions with priorities. For each, \( T_r \in S \),

(i) \( T_r \subseteq T \), and \( |T_r| > 1 \);

(ii) \( \forall t, t_j \in T_r, (j > i) \), transition \( t \) and \( t_j \) could be enabled at the same time, and have different priorities. The priorities of \( t \) in group \( T_r \) is \( P_r(t) \), and \( t_j \) is

\[
P_r(t_j) \geq P_r(t) ;
\]

(iii) In group of transitions with priorities \( T_r = \{ t_i, t_2, \ldots \} \), at least one \( t \), making \( P_r(t) > P_r(t_i) \).
3.3 \( \ast t_i \) (Absolute Preset of Transition \( t_i \))

Absolute preset of transition \( t_i \) is composed of the places, which are elements of preset of \( t_i \), but not be preset of other transitions in the same group \( T_p \).

\[
\{ \ast t_i \} = \{ p \mid p \in \ast t_i \land (\forall t_k \in T_p (k \neq i), p \notin \ast t_k) \}.
\] (1)

3.4 \( P_k \) (Key Place Set)

\( P_k \) (Key Place Set) is a set of places, effecting transition priorities in same group. Search algorithm of \( P_k \) as follows:

STEP 1: Identify Group of Transitions with Priorities \( T_p \).

STEP 2: Find Key Place Set \( P_k \) according to the following Equation:

\[
P_k = \bigcup_{i=1}^{n} \ast t_i - \bigcap_{i=1}^{n} t_i | t_i \in T_p \} - \{ \ast t_k \}.
\] (2)

Firstly, identify preset for each transition \( t_i \), \( t_i \in T_p \) in group \( T_p \), then, removed public preset of them. Finally, expurgated the absolute preset of transition \( t_i \) (with the lowest priority in group \( T_p \)).

3.5 \( C_i(p) \) (Key Colour Set of Key Place \( p)\) and \( C_j(P_k) \) (Key Colour Set of Key Place Set \( P_k \))

\( C_i(p) \) (Key Colour Set of Key Place \( p \) for short) Key Colour Set is a multi-set of colour set of key place \( p(p \in P_k) \). They effected the change of tokens in the key place \( p \).

Usually, if the key place \( p \) has simple colours set \( C(p) \), the key colour set of \( p \) is same with \( C(p) \). Otherwise, \( p \) has compound colour set, the key colour set is a multi-set, composed with elements in compound colour set.

\( C_j(P_k) \) (Key Colour Set of Key Place Set \( P_k \)) is a multi-set of key colour set for each key place \( p \) in group \( P_k \).

\[
C_j(P_k) = \sum_{k=1}^{n} C_j(p).
\] (3)

Supposing that:

(i) The key place set is \( P_k = \{ p_1, p_2 \} \).

(ii) The colour set of \( p_1 \), a simple colour set, is \( C(p_1) = c_1 \), and the key colour set of \( p_1 \) is \( C_j(p_1) = c_1 \).

(iii) The colour set of \( p_2 \), a compound colour set \( C(p_2) = c_1 \cdot c_2 \cdot c_3 \) and the key colour set of \( p_2 \) is \( C_j(p_2) = \{ c_1, c_2 \} \).

Then, \( C_j(P_k) = \{ c_1, c_1, c_2 \} \) is a multi-set, and the length of \( C_j(P_k) \) is \( |C_j(P_k)| = 3 \).

3.6 \( \ast p \) (Relative Preset of \( P \)) and \( \ast p \) (Relative Postset of \( P \))

\( p \) is a key place. The relative preset and postset of \( p \) must be set of transitions.

\[
\ast p = \{ t \mid t \in \ast p \land (Type(E(a_m)) \cap C_j(p) \neq \emptyset) \},
\] (4)

\[
\ast p = \{ t \mid t \in \ast p \land (Type(E(a_m)) \cap C_j(p) \neq \emptyset) \}.
\] (5)

4 Algorithms of Transitions Priorities in Group

4.1 Algorithm 1: Model for \( |P_k| = 1 \) and \( |C_j(P_k)| = 1 \)

Example 1: FIGURE 2 is a typical of concurrent systems with priority relationships. When \( t_1 \) and \( t_2 \) are enabled at the same time, \( t_1 \) has a higher priority. However, when \( t_{10} \) and \( t_1 \) are enabled simultaneously, they have no priorities, randomly trigger.

FIGURE 2 Source Model, with \( |P_k| = 1 \) and \( |C_j(P_k)| = 1 \)

STEP 1: Identify group of transitions with priorities \( T_p \).

In FIGURE 2, transition set is \( T = \{ t_1, t_2, t_{10} \} \). \( t_1 \) and \( t_2 \) have priorities between them, but have no priorities with other transitions. Furthermore, the priority of \( t_1 \) is higher than \( t_2 \). The group of enabled transitions with priorities is \( T_p \), \( T_p = \{ t_1, t_2 \} \) and \( P_{t_1}(t_1) > P_{t_2}(t_2) \).
STEP 2: Identify $P_i$, the key place set of $T_p$, and $C_i(P_i)$, the key colour set of $P_i$.

\[ t_i = \{ p_{i1}, p_{i2} \} \]

\[ t_2 = \{ p_{i2} \} \]

So, $P_i = \bigcup_{i=1}^{2} t_i \setminus \bigcap_{i=1}^{2} t_i = \{ p_{i1} \}$.

$C_i(p_{i1}) = \{ \text{PROCESS} \}$,

$C_i(p_{i2}) = \{ \text{PROCESS} \}$.

STEP 3: Modify the CPN model, increasing one “counter place” to calculate tokens in key place set.

STEP 3.1: Increase variable for “counter place”.

\[ \text{var} \ n: \text{INT}; \]

STEP 3.2: Increase “counter place” for key place set $P_k = \{ p_{k1} \}$ in CP-net model, with definition of colour set and initial value.

Increase “counter place” count, with colorset INT. Because the initial count of the key colour PROCESS is 0, the initial value of “counter place” count is 0.

STEP 3.3: If the relative preset of the key place $p$ is not empty, add two arcs from “counter place” count to $p$; and if the relative postset of the key place $p$ is not empty, add two arcs from “counter place” count to $\overline{p}$.

Added arcs describe the change of tokens in key place.

Transition $t_{10}$ triggered, place $p_{i1}$. Increase one PROCESS, while $t_1$ triggered, place $p_{i1}$ decreased one PROCESS. So, added two arcs from count to its relative preset $\overline{p_{i1}} = \{ t_{10} \}$, one is outgoing arc with function “n”, the other is incoming arc with function “n+1”. Similarly, added two arcs from count to its relative postset $\overline{p_{i1}} = \{ t_1 \}$, one is outgoing arc with function “n”, the other is incoming arc with function “n-1”.

STEP 3.4: added one bi-directional arc with function “0”, from “counter place” count to other transitions with lower priorities in the same group.

For $T_p = \{ t_1, t_2 \}$ in the model, extracted one bi-directional arc from count to $t_2$.

The finished model is shown in Figure 3.

4.2 ALGORITHM 2: MODEL FOR $|P_i| \geq 1$ AND $|C_i(P_i)| > 1$

Example 2: Added two nodes, $p_{21}$ and $t_{20}$, in FIGURE 2. Transition $t_1$ could enable, only when $p_{i1}$, $p_{i2}$ and $p_{i3}$ have tokens in place. Furthermore, $t_1$ has a higher priority than $t_2$. However, $t_1$ and $t_2$ couldn’t plunder privilege of other transitions, $t_1$ should have same priority with $t_{i0}$ and $t_{20}$, as shown in FIGURE 4.

FIGURE 4 Source Model with $|P_i| = 1$ and $|C_i(P_i)| > 1$

STEP 1: Identify group of transitions with priorities $T_p$.

In Example 2, transition set is $T = \{ t_1, t_2, t_{i0}, t_{20} \}$. The priority of $t_1$ is higher than $t_2$. The group of transitions with priorities is $T_p$, $T_p = \{ t_1, t_2 \}$, and $P_{t_1}(t_1) > P_{t_2}(t_2)$.

STEP 2: Identify $P_i$, the key place set of $T_p$, and $C_i(P_i)$, the key colour set of $P_i$.

\[ t_1 = \{ p_{i1}, p_{i2}, p_0 \} \]

\[ t_2 = \{ p_{i3} \} \]

$P_i = t_1 \cup t_2 = t_1 \cap t_2 = \{ p_{i1}, p_{i2}, p_{i3} \} \setminus \{ p_{i1} \} = \{ p_{i1}, p_{i2}, p_{i3} \} - \{ p_{i1} \} = \Phi = \{ p_{i1}, p_{i2}, p_{i3} \}$

Furthermore:

$C_i(p_{i1}) = \{ \text{PROCESS} \}$,

$C_i(p_{i2}) = \{ \text{PROCESS} \}$.

So: $C_i(P_i) = \{ \text{PROCESS, PROCESS} \}$.

STEP 3: Modify the CPN model. Increase a “counter place” to calculate tokens in key place set.

STEP 3.1: Increase variables for “counter place” and trigger function for transition with lower priorities.

According to $C_i(P_i) = \{ \text{PROCESS, PROCESS} \}$ and $|C_i(P_i)| = 2$, added a “counter place” with 2-tuples in the model for $p_{i1}$ and $p_{i2}$. Defined variables and trigger function by CPN ML as follows:

//new colorset for "counter place" with two-tuples

```
colset INTLIST = product INT * INT;
var n1,n2:INT;
fun tran(counts:INTLIST) =
```
STEP 3.2: Increase “counter place” for key place \( P_k = \{p_{11}, p_{22}\} \) in CP-net model, with definition of colour set and initial value.

Increase “counter place” \( count \), with colour set \( \text{INTLIST} \). Because the initial count of the key colour \( \text{PROCCESS} \) in \( p_{11} \) and \( p_{22} \) are all 0, the initial value of “counter place” \( count \) is \((0,0)\).

**FIGURE 5** Model with priorities with \(|P| \geq 1\) and \(|C(P)| > 1\).

STEP 3.3: Do as follows for each \( \forall p \in P \): if the relative preset of the key place \( p \) is not empty, add two arcs from “counter place” \( count \) to \( p \); and if the relative postset of the key place \( p \) is not empty, add two arcs from “counter place” \( count \) to \( p \). Added arcs describe the change of tokens in key place \( Count \), the “counter place”, looks like the format \( (n_1, n_2) \). And \( n_1 \) showed the change of \( p_{11} \), \( n_2 \) records the change of \( p_{22} \). Furthermore, changed in adding arcs and the arc functions (as shown in Figure 5).

STEP 3.4: If there is no outcoming arc from “counter place” \( count \) to other transitions with lower priorities in same group, added one bi-directional arc with function \( (n_1, n_2) \).

STEP 3.5: Added trigger function in all transitions with lower priorities.

In this example, added trigger function \( tran(n_1, n_2) \) for transition \( t_1 \).

5 Practise example

Example 3: The systems have a main program Process, and two external interrupts \( \text{INTR}_1 \) and \( \text{INTR}_2 \), which have the priorities relationships that the priorities of \( \text{INTR}_1 \) is higher than \( \text{INTR}_2 \), namely \( P_{\text{INTR}_1} > P_{\text{INTR}_2} \). The question is how to construct a CPN model to meet interrupts priorities.

The main process \( \text{Process} \) set interrupts on or off. The place \( \text{INTR} \) managed all the interrupts that is, when \( \text{Process} \) set the interrupt on, it send semaphore \( \text{intr} \) to transition \( \text{on}\_\text{intr} \) in order to enable the active interrupt. Otherwise, when it set the interrupt off, it remove \( \text{intr} \) from transition \( \text{on}\_\text{intr} \), thus disabled the interrupt, as shown in Figure 6.

**FIGURE 6** Interrupt model \( M_1 \) without priorities

The place \( \text{START\_\text{INTR}} \) simulated the external interrupts, the transition \( \text{random} \) control the happening of interrupts randomly, and the place \( \text{Out} \) represented the current enabled interrupts. The interrupt conditions are:

(i) \( \text{INTR}_1 \) had been enabled;

(ii) The existence of the external interrupt \( \text{INTR}_2 \), then \( \text{Interrupts} \) could run. However, the model could be shown as **FIGURE 6**, although given the current disconnect, turn control to an external interrupt occurs, but not showing the priorities of relations between \( \text{INTR}_1 \) and \( \text{INTR}_2 \).
If omitted other unrelated contents, the interrupt model could be minimized, as shown in Figure 7.

The minimized interrupt Model \( M_4 \) use the above algorithm, as show as Section IV.

STEP 1: Identify group of transitions with priorities \( T_p \).

In Figure 7, transition set is \( T = \{ \text{random, Interrupt}_1, \text{Interrupt}_2 \} \). The group of enabled transitions with priorities is \( T_p \).

\[ T_p = \{ \text{Interrupt}_1, \text{Interrupt}_2 \} \]

So, \( P_r (\text{Interrupt}_1) > P_r (\text{Interrupt}_2) \).

STEP 2: Identify \( P_r \), the key place set of \( T_p \), and \( C_k (P_r) \), the key colour set of \( P_r \).

\* \( \text{Interrupt}_1 = \{ \text{out} \} \)
\* \( \text{Interrupt}_2 = \{ \text{out} \} \)

Thus,

\[ P_r = \{ \text{Interrupt}_1 \cup \text{Interrupt}_2 \} = \Phi \]

\* \( \text{Interrupt}_1 \cap \text{Interrupt}_2 = \{ \text{out} \} \)

Namely, it became an unsolvable problem by using the above algorithm.

Secondary transitions, \( \text{call intr}_1, \text{call intr}_2 \), and places, \( \text{OUTING}_1, \text{OUTING}_2 \), could be introduced to the previous model \( M_4 \). Furthermore, place \( \text{OUT} \) would be divided into two places \( \text{OUT}_1 \) and \( \text{OUT}_2 \). Thus, the model \( M_4 \) could be changed as \( M_4' \), shown in Figure 8.

FIGURE 7 Minimized interrupt model \( M_4 \)

FIGURE 8 Auxiliary Model \( M_4' \)

\* \( \text{Interrupt}_1 = \{ \text{OUTING}_1 \} \)
\* \( \text{Interrupt}_2 = \{ \text{OUTING}_2 \} \)

Thus,

\[ P_r = \{ \text{Interrupt}_1 \cup \text{Interrupt}_2 \} = \{ \text{OUTING}_1, \text{OUTING}_2 \} \]

Next,

\[ C_r (\text{OUTING}_1) = \{ \text{intr} \}, C_r (\text{OUTING}_2) = \{ \text{intr} \} \]

In the model \( M_4' \), when the places, \( \text{OUTING}_1 \), and \( \text{OUTING}_2 \), being changed, they could affect trigger priorities of transitions \( \text{Interrupt}_1 \) and \( \text{Interrupt}_2 \).

STEP 3: Modify the CPN model, increasing one “counter place” to calculate tokens in key place set, as shown in Figure 9.

FIGURE 9 Partial interrupt model with priorities

Add the modified partial model to the original model to form a complete interrupt model with priorities \( M_5 \), as shown in Figure 10.

In the complete model, the main process \( \text{Process} \) could send semaphore to the place \( \text{INTR} \) to control whether the interrupts are enabled. Moreover, the transition \( \text{Interrupt}_1 \) or \( \text{Interrupt}_2 \) could be tangled only when outing interrupts signal is enabled. Meantime, only the transition \( \text{Interrupt}_1 \) could be triggered although it and \( \text{Interrupt}_2 \) could be enabled simultaneously, namely, \( \text{Interrupt}_1 \) have higher priorities than \( \text{Interrupt}_2 \).
6 Working outlook

This paper researched the CPN model with transition priorities in Group. When the original model has lower complexity, that was, \(|P|\) and \(|C(P)\) is small, algorithm for transition with priorities is better. However, this algorithm only addressed transition priorities under enabling condition, not under certain conditions. After that, it could expand the research work in this regard. Furthermore, due to “count place” need to connect bi-directional arc to multiple transitions, making the more complex model structure, reducing the complexity of the model is also the focus of future research.

References

### Authors

**Hong Wang**, born in March, 1978, Tieling, Liaoning Province, China  
*Current position, grades:* PhD candidate at Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, a lecturer in the School of Control and Computer Engineering, North China Electric Power University.  
*University studies:* computer science technology and application.  
*Scientific interest:* high reliability software testing and model checking.  
*Publications:* 3 papers.

**Tao Zhang**, born in December, 1972, Zhaozhuang, Shandong Province, China  
*Current position, grades:* professor at Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences.  
*University studies:* PhD at Tsinghua University in China.  
*Scientific interest:* high reliability software testing and validation, high reliability electronic information system analysis, design methods, complex system simulation.  
*Publications:* more than 20 papers.