
(Murata’s paper is available in Proc. IEEE, April 1989)


A modeling tool for system exhibiting

- Concurrency
- Synchronization
- Randomness

Used in Performance evaluation, high level representation and in Modeling.

Important varieties:

\[ PN \rightarrow (SPN, TPN, GSPN, CPN, HLPN, APN) \]

SPN generates stochastic processes, a powerful computational paradigm like Markov Systems

TPN is Timed PN

GSPN, a generalized stochastic PN

CPN, colored PN

HLPN, High level PN

ABN, an Abstract PN

A simple PN is a 5-tuple: \((P, T, IN, OUT, M_0)\)

\(P\), finite number of Places: \(\{p_1, p_2, \ldots, p_n\}\)
$T$, finite set of transitions: \{ $t_1, t_2, \ldots, t_m$ \}

$P \cup T \neq \emptyset \quad P \cap T = \emptyset$

$IN : (P \times T) \rightarrow N$ input function. Defines directed arcs from places to transitions.

$OUT : (T \times P) \rightarrow N$ output function. Defines directed arcs from transition to places

$M_0 : P \rightarrow N$ initial marking

Example.

Other examples of Petri Nets.

A Production Net:

From http://www.scholarpedia.org/article/Petri_net
An organization Net.

From http://www.scholarpedia.org/article/Petri_net

Marked Petri Nets

\[ M_0 = (m_0_1, m_0_2, ..., m_0_n) \], Initial marking. An example.

In this case, \( M_0 = (10010) \)

Execution rules

- A PN executes by firing transactions
- A transition may be fired if enabled
A transition is enabled when all of its input places contain at least one token

Example.
a. output places to a fired transition

\[ m_p^{new} = m_p^{old} + 1 \]

b. Input places to a fired transition

Extensions to PN

Multiple arcs

\[ N( p_i, t_j ) = \text{Number of arcs from } p_i \text{ to } t_j \]
\[ N(t_j, p_i) = \text{Number of arcs from } t_j \text{ to } p_i \]

Inhibitor arcs

A transition is enabled when all of its normal input places contain at least one token and no token.

A behavioral profile of a ATM process:

PN of a ATM Process

This is a typical sequential process. No concurrency is required.

Concurrency and conflict.
This PN shows conflict and concurrency.

Producer/Consumer problem. Infinite buffer solution.

With \( m \) producers and \( n \) consumers, we have the following
Various attributes of PN

- Safeness: Number of tokens in each place cannot exceed one
- Boundedness: The number of tokens in each place cannot exceed some threshold $k$
- Conservation: Total number of tokens in the system is invariant.
- Dead transition: A transition that can never be fired in future.
- Live Transitions: transitions that can be enabled
- Deadlock: No transition can fire
Live PN: Every transition is live
Reachable markings: A marking $M$ obtained by firing a sequence of transitions from initial marking $M_0$.

Example.

$$M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow$$

Reachability Tree

Example.

The partial Reachability tree is
Marking graph algebra. Define $A(i, j) = I(p_i)$ if place $p_i \rightarrow t_j$ (incident on). $T(p_i) =$ Number of token in the input place $p_i$. This is the input incidence matrix.

Similarly, define the output incidence matrix $O(i, j)$ as $O(i, j) = T(p_j)$ if the transition $t_i \rightarrow p_j$. This is the Output Incidence matrix. Define the joint incidence matrix $A(i, j)$ as

$$A(i, j) = O(i, j) - I(i, j)$$

The necessary condition for marking $M$ to be reachable from marking $M_0$ is that there must exist a firing vector $v$ such that
\[ M = M_0 + Av \]

Example. For the Petri Net

![Petri Net Diagram]

The joint incidence matrix is

\[
\begin{pmatrix}
-1 & 0 & 0 \\
1 & 1 & -1 \\
0 & -1 & 1
\end{pmatrix}
\]

A PN is T-invariant if there exists a sequence of firings that brings a marking back to whence it started. Cyclic Schedules correspond to T-invariant PN.