Petri Nets: Tutorial and Applications

Jeffrey W. Herrmann
Edward Lin

November 5, 1997

The 32th Annual Symposium of the Washington Operations Research -
Management Science Council
Washington, D.C.

CIM Lab
Institute for Systems Research
University of Maryland
College Park, Maryland
Outline

● Purpose
● Applications
● What is a Petri Net?
● Dynamics
● Basic Constructs
● Properties
● Analysis Methods
● Extensions of Petri Nets
● Resources for Petri Nets
● Summary
Purpose

- To describe the fundamentals of Petri nets so that you begin to understand what they are and how they are used.
- To give you resources that you can use to learn more about Petri nets.
Petri Net Applications

- Manufacturing, production, and scheduling systems
- Sequence controllers (Programmable Logic Controller, PLC)
- Communication protocols and networks
- Software -- design, specification, simulation, validation, and implementation
Petri Nets -- Graphic Tool

- A bipartite directed graph containing places (circles), transitions (bars), and directed arcs (places <-> transitions).

- Places -- buffers, locations, states
- Transitions -- events, actions
- Tokens -- parts
A Petri net is a four-tuple:

$$PN = <P, T, I, O>$$

* $P$: a finite set of places, $\{p_1, p_2, \ldots, p_n\}$
* $T$: a finite set of transitions, $\{t_1, t_2, \ldots, t_s\}$
* $I$: an input function, $(T \times P) \rightarrow \{0, 1\}$
* $O$: an output function, $(T \times P) \rightarrow \{0, 1\}$

$M^0$: an initial marking, $P \rightarrow N$

$$<P, T, I, O, M^0>$$ -- a marked Petri net
An Example

- $P = \{p1, p2, p3\}$
- $T = \{t1, t2, t3\}$
- $I = \begin{bmatrix} p1 & p2 & p3 \\ t1 & 1 & 0 & 0 \\ t2 & 0 & 1 & 0 \\ t3 & 0 & 0 & 1 \end{bmatrix}$
- $O = \begin{bmatrix} p1 & p2 & p3 \\ t1 & 0 & 1 & 0 \\ t2 & 0 & 0 & 1 \\ t3 & 1 & 0 & 0 \end{bmatrix}$
- $M^0 = (1, 0, 0)$

Note:
- $p1$ is the input place of transition $t1$
- $p2$ is the output place of transition $t1$
Dynamics

- **Enabling Rule:**
  - A transition $t$ is enabled if every input place contains at least one token

- **Firing Rule:**
  - Firing an enabled transition
    - removes one token from each input place of the transition
    - adds one token to each output place of the transition
Dynamics

Initial State:

State after $t_1$ is fired:

State after $t_2$ is fired:

State after $t_3$ is fired:
Basic Constructs

- Sequential actions
- Dependency
- Conflict (decision, choice)
- Concurrency
- Cycles
- Synchronization - (mutually exclusive actions, resource sharing, communication, queues)
Sequential Actions

Each action is a transition.
Dependency

A transition requires two inputs.
Conflict Construct

Only one of the two transitions can fire.
Concurrency Construct

These two sequences can occur simultaneously.
Cycles

loading  processing  Unloading

Edward Lin, University of Maryland
Synchronization

Machine can process one part at once.
Resource Sharing

One worker for two machines. The worker can work at one machine at a time.
Buffer (Queue)

The buffer can hold a limited number of parts.
Communication

Program 1

Program 2
An Example

Machine States:
- Loading
- Processing
- Waiting for unloading
- Unloading

Buffer State:
- Space availability
Put It Together

Flowchart showing the process of loading, processing, waiting for unloading, and unloading. The flowchart includes two machines labeled Machine 1 and Machine 2, with a robot and a buffer available as part of the system. The diagram illustrates the flow of materials through the system.
<table>
<thead>
<tr>
<th>Property</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundedness</td>
<td>Work-in-process</td>
</tr>
<tr>
<td>- the number of</td>
<td></td>
</tr>
<tr>
<td>tokens in a</td>
<td></td>
</tr>
<tr>
<td>place is</td>
<td></td>
</tr>
<tr>
<td>bounded</td>
<td></td>
</tr>
<tr>
<td>Safeness</td>
<td>Hardware devices</td>
</tr>
<tr>
<td>- the number of</td>
<td></td>
</tr>
<tr>
<td>tokens in a</td>
<td></td>
</tr>
<tr>
<td>place never</td>
<td></td>
</tr>
<tr>
<td>exceeds one</td>
<td></td>
</tr>
<tr>
<td>Deadlock-free</td>
<td>Resources competing</td>
</tr>
<tr>
<td>- none of</td>
<td></td>
</tr>
<tr>
<td>markings in</td>
<td></td>
</tr>
<tr>
<td>$R(PN, M^0)$ is</td>
<td></td>
</tr>
<tr>
<td>a deadlock</td>
<td></td>
</tr>
<tr>
<td>Reachability</td>
<td>Messages delivery</td>
</tr>
<tr>
<td>- find $R(PN, M^0)$</td>
<td></td>
</tr>
</tbody>
</table>
Analysis Methods

- Enumeration
  - Reachability Tree
  - Coverability Tree

- Linear Algebraic Technique
  - State Matrix Equation
  - Invariant Analysis: P-Invariant and T-invariant

- Simulation
Reachability Tree (1)

Initialization: \[ M_0 = (1,0,0,0,0,1) \]

Step 1:
\[ M_0 = (1,0,0,0,0,1) \]
\[ M_1 = (0,1,0,1,0,1) \]

Step 2:
\[ M_0 = (1,0,0,0,0,1) \]
\[ M_1 = (0,1,0,1,0,1) \]
\[ M_2 = (0,0,1,1,0,1) \]
\[ M_4 = (0,1,0,0,1,1) \]
Reachability Tree (2)

Step 3:

- \( M_0 = (1,0,0,0,0,1) \)
- \( M_1 = (0,1,0,1,0,1) \)
- \( M_2 = (0,0,1,1,0,1) \)
- \( M_4 = (0,1,0,0,1,1) \)
- \( M_3 = (0,0,1,0,1,1) \)

Step 4:

- \( M_0 = (1,0,0,0,0,1) \)
- \( M_1 = (0,1,0,1,0,1) \)
- \( M_2 = (0,0,1,1,0,1) \)
- \( M_4 = (0,1,0,0,1,1) \)
- \( M_3 = (0,0,1,0,1,1) \)
Reachability Tree (3)

Step 5:

- $M_0 = (1, 0, 0, 0, 0, 1)$
- $M_1 = (0, 1, 0, 1, 0, 1)$
- $M_2 = (0, 0, 1, 1, 0, 1)$
- $M_3 = (0, 0, 1, 0, 1, 1)$
- $M_4 = (0, 1, 0, 0, 1, 1)$

Diagram:

- $t_1$ from $p_1$ to $p_2$ and $p_4$
- $t_2$ from $p_3$ to $p_6$ and $t_3$
- $t_4$ from $p_3$ to $M_3$
Reachability Tree/Graph

\[ M_0 = (1,0,0,0,0,1) \]
\[ M_1 = (0,1,0,1,0,1) \]
\[ M_2 = (0,0,1,1,0,1) \]
\[ M_3 = (0,0,1,0,1,1) \]
\[ M_4 = (0,1,0,0,1,1) \]
Reachability Tree
Coverability Tree (1)

Initialization: \[ M_0 = (1,0,0) \] new

Step 1:
\[ t_1 \xrightarrow{(1,0,0)} t_2 \]
\[ m_1 = (1,1,0) \geq (1,0,0) \]
\[ \text{new } m_1 = (1,\omega,0) \quad m_2 = (0,1,1) \text{ new} \]

Step 2 (m1):
\[ t_1 \xrightarrow{(1,0,0)} t_2 \]
\[ \text{new } m_1 = (1,\omega,0) \quad m_2 = (0,1,1) \text{ new} \]
\[ t_1 \quad t_2 \]
\[ \text{old } (1,\omega,0) \quad \text{new } m_3 = (0,\omega,1) \]
Step 3 (m3):

\[ (1,0,0) \text{ new} \]

\[ t_1 \quad t_2 \]

new \( m_1 = (1, \omega, 0) \)

\[ m_2 = (0,1,1) \text{ new} \]

\[ t_1 \quad t_2 \]

old \( (1, \omega, 0) \)

\[ m_3 = (0, \omega, 1) \text{ new} \]

\[ \quad t_3 \]

old \( (0, \omega, 1) \)

Step 4 (m2):

\[ (1,0,0) \text{ new} \]

\[ t_1 \quad t_2 \]

new \( m_1 = (1, \omega, 0) \)

\[ m_2 = (0,1,1) \text{ new} \]

\[ \quad t_3 \]

old \( (0, \omega, 1) \)

\[ m_4 = (0,0,1) \text{ new} \]
Step 5 (m4): Coverability Tree

Reachability Tree

Edward Lin, University of Maryland
Linear Algebraic Technique

State Equation: $M = M^0 + \mu A$, where $\mu$ is a vector with $s$ elements

- $O = \begin{bmatrix} p1 & p2 & p3 \\ t1 & 0 & 1 & 0 \\ t2 & 0 & 0 & 1 \\ t3 & 1 & 0 & 0 \end{bmatrix}$
- $I = \begin{bmatrix} p1 & p2 & p3 \\ t1 & 1 & 0 & 0 \\ t2 & 0 & 1 & 0 \\ t3 & 0 & 0 & 1 \end{bmatrix}$
- $M^0 = (1, 0, 0)$

Incidence Matrix
- $A = O - I$

$$A = \begin{bmatrix} p1 & p2 & p3 \\ t1 & -1 & 1 & 0 \\ t2 & 0 & -1 & 1 \\ t3 & 1 & 0 & -1 \end{bmatrix}$$
Linear Algebraic Technique

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
-1 & 1 & 0 \\
0 & -1 & 1 \\
1 & 0 & -1
\end{bmatrix}
\]

t1 fired

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
-1 & 1 & 0 \\
0 & -1 & 1 \\
1 & 0 & -1
\end{bmatrix}
\]

t1, t2 fired

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
-1 & 1 & 0 \\
0 & -1 & 1 \\
1 & 0 & -1
\end{bmatrix}
\]

t1 t2 t3 fired
T-Invariant

T-Invariant: $YA = 0$, where $Y$ is a s element vector
$Y$ is the number of transition firings

\[
\begin{bmatrix}
y_1 & y_2 & y_3
\end{bmatrix}
\begin{bmatrix}
-1 & 1 & 0 \\
0 & -1 & 1 \\
1 & 0 & -1
\end{bmatrix}
= 0
\]

$-y_1 + y_3 = 0$
$y_1 - y_2 = 0$
$y_2 - y_3 = 0$

$y_1 = y_2 = y_3$

minimum t-invariant = (1, 1, 1)
T-Invariant

(1,0,0)
\downarrow
(0,1,0)
\downarrow
(0,0,1)
\downarrow
(1,0,0)

(1,1,1)

(1,0,0) \rightarrow (1,0,0)
P-Invariant

P-Invariant: $AX^T = 0$, where $X$ is a $n$ element vector, $X$ is the weight of each place

\[
\begin{bmatrix}
-1 & 1 & 0 \\
0 & -1 & 1 \\
1 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} = 0
\]

\begin{align*}
-x_1 + x_2 &= 0 \\
-x_2 + x_3 &= 0 \\
x_1 - x_3 &= 0
\end{align*}

minimum $p$-invariant = $(1, 1, 1)$

The quantity $S = x_1 M(p1) + x_2 M(p2) + x_3 M(p3)$
The quantity $S = x_1 M(p_1) + x_2 M(p_2) + x_3 M(p_3)$

$1 = 1 M(p_1) + 1 M(p_2) + 1 M(p_3)$
Simulation

● Discrete event simulation
● Same model for simulation and analysis
● Need rules to resolve conflicts
● Useful for validation and visualization
Extensions of Petri Nets

- **Event Graph (marked graph, decision-free)**
  - Each place has exactly one input transition and exactly one output transition

- **Deterministic Timed Petri Nets**
  - Deterministic time delays with transitions

- **Stochastic Timed Petri Nets**
  - Stochastic time delays with transitions

- **Color Petri Nets**
  - Tokens with different colors

- **Hybrid Nets**
  - Combine object-oriented concept into Petri nets
Further Readings

- Petri nets home page: http://www.daimi.aau.dk/%7Epetrinet/
- Petri nets mailing list: PetriNets@daimi.aau.dk
- Coloured Petri nets: http://www.daimi.aau.dk/designCPN/
- Petri nets standard: http://www.daimi.aau.dk/%7Epetrinet/standard/


- Computer Integrated Laboratory(CIM Lab) page: http://www.isr.umd.edu/Labs/CIM/
Summary

- A graphical and mathematical tool
- Applications
- Constructs
- Properties: Boundedness, Safeness, Deadlock-free, liveness, Reachability
- Analysis Techniques:
  - Reachability trees
  - Coverability trees
  - Linear algebraic techniques
  - Simulation
- Extensions
- Resources