

Introduction

Nowadays, computer systems have become very complex. In addition, dynamic systems cannot be described by only taking into account their initial and final states. Indeed, we must take into account their permanent behavior which is a sequence of states, so that we can speak of a well-founded description. Among the large number of formal techniques that have already been proposed to specify, analyze and verify this kind of systems, Petri nets are one of the most widely used.

Petri nets offer a formal tool (Semantics and Mathematics) and a good graphical representation which make it possible to model and analyze discrete systems, particularly concurrent and parallel systems. The graphic aspect of the Petri nets helps us easily understand the modeled system. Furthermore, their power of mathematical expression makes it possible to simulate dynamic and competing activities. The major interest of these networks lies in their possibility of analyzing the modeled systems, thanks to graph models and algebraic equations (mathematical aspect). Petri nets are modeling tools generally used in the preliminary phase of system design in order to carry out their functional specification, modeling and monitor their evaluation. Thanks to their expressiveness and flexibility, they are used in a wide variety of fields. They allow in particular: The modeling of computer systems, the evaluation of the performance of discrete systems, human-machine interfaces, the control of manufacturing workshops, the design of real-time systems, the modeling of communication protocols and the modeling of chains production (manufacturing), distributed systems and computer architecture, etc.

1. Definition of PN

A Petri Net (PN) is a graph composed of two kinds of nodes:

-  **Places:** (represented by circles) which allow the description of the possible states (conditions) of the system. 
-  **Transitions:** (represented by bars) which allow the description of the events or the actions which cause the change of state. 
-  Satisfaction of a condition: modeled using a token. 
-  Condition true 
-  Condition false 

Petri Nets offer a description of a system by making a clear distinction between the static aspect and the dynamic aspect, such as its easy and understandable graphical representation by users. The analysis of a Petri net can state important characteristics of the system regarding its structure and dynamic behavior. The results of this analysis are used to evaluate the system and enable modification or improvement. The following figure represents the general modeling method:

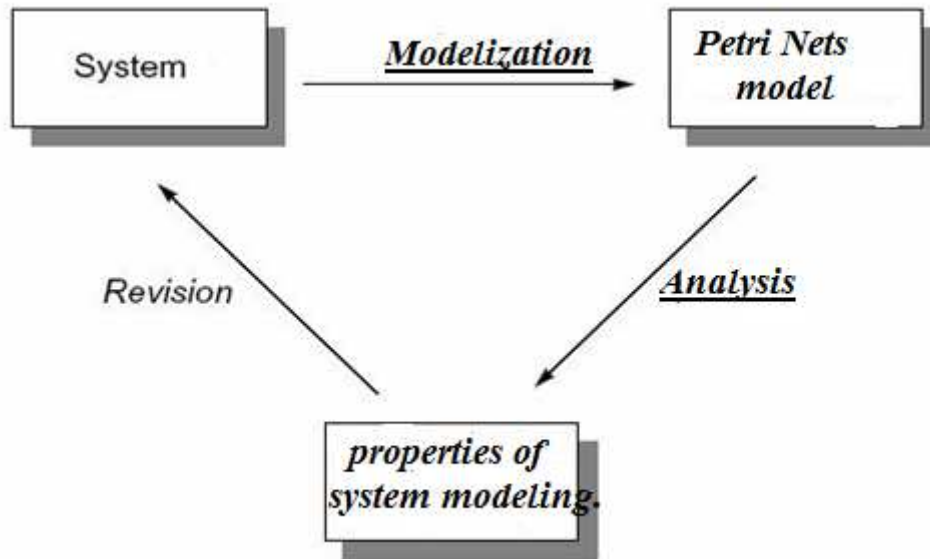


Figure 01: General modeling and analysis method based on Petri nets.

- **Basic concepts for PNs :** There are three basic concepts: [BENMOUSSA, HOUIDI ; 2015]
- **Condition:** A condition is a predicate or logical description of a state of the system. A condition can be true or false. A system state can be described as a set of conditions.
 - **Event:** Events are actions taking place in the system. Firing an event depends on the system state
 - **Firing, pre-condition, post-condition:** The conditions necessary to fire an event are the pre-conditions of the event. When an event occurs, some of its pre-conditions may cease to be true while other conditions, called post-conditions of the event, become true.

1.1. Structural aspect

1.1.1. Formally

A Petri net is defined by a quadruplet $PN = (P, T, E, S)$ Or:

- ➡ $P = (p_1, p_2 \dots p_m)$, a finite set of places.
- ➡ $T = (t_1, t_2 \dots t_m)$, a finite set of transitions.
- ➡ E the forward incidence function, connect transitions to places, $E: P * T \rightarrow N$
- ➡ S the backward incidence function, connect places to transitions, $S: P * T \rightarrow N$

The place set and the transition set are disjoint $P \cap T = \emptyset$ [Bah05]

1.1.2. Graphic Representation

A Petri net is represented by a directed graph composed of arcs connecting places and transitions. Two places cannot be connected together, nor two transitions. [Wiki]. Places may contain tokens. The distribution of tokens in the places is called the marking of the Petri net. [Boimond04]. Places: are represented by circles which contain tokens. And the transitions: are represented by rectangles and represent events.

- There are two types of transition: a place to a transition or a transition to a place and the following figure shows us this notion:

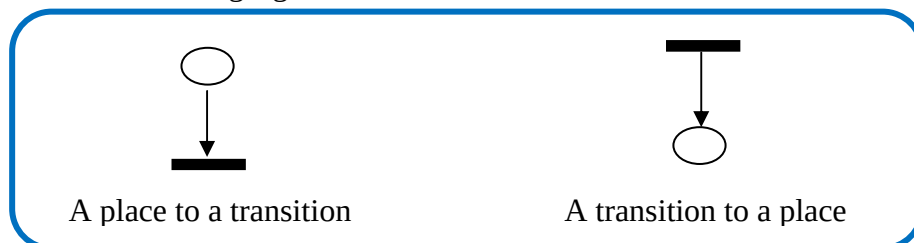


Figure 02: The different types of transition.

Firing a transition leads to the change of state of the network. This firing corresponds to token movements from an entry place to an exit place of the transition.

The following figure represents an example “Model of operations on two natural integers”: Let n_1 and n_2 be two natural integers. We want to model the addition of these two integers by a PN.

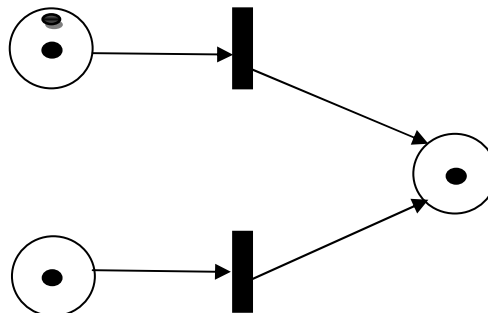


Figure 03: Addition of 2 integers using a Petri net.

The state of the system is characterized by:

- ◆ The number n_1 at the initial moment (number of marks in place P1)
- ◆ The number n_2 at the initial moment (number of marks in place P2)
- ◆ The sum $n_1 + n_2$ at the final instant (number of marks in place P3)

The state of the system will evolve when:

- ◆ A mark is removed from P1 to be placed in P3 (The firing of an enabled transition T1)
- ◆ A mark is removed from P2 to be placed in P3 (The firing of an enabled transition T2)

1.2. Behavioral aspect

1.2.1. Marking

A marked Petri net is defined by the pair $\{N, M\}$, where N is a Petri net, and $M: P \rightarrow \mathbb{N}$ is an application called the marking function which associates with each place in N a number of tokens. The state of a network is often called network marking.

A labeled Petri net is given by $\Sigma = (P, T, W, M_0)$ where M_0 is the initial marking.

The behavior of a marked Petri net is defined by a firing rule.

- ➡ Concept of change of state (firing rule): A change of state is denoted by a movement of tokens (black dots) from entry places to exit places of a transition. This movement is caused by firing a transition **[RDP]**. Firing is conditioned by the presence of tokens in the entry places of the transition; each place contains a positive or zero integer number of marks or tokens. Below is an example of marking:

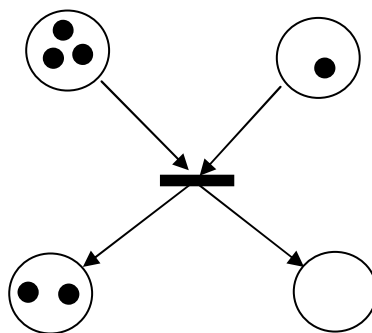


Figure 04: Marking.

1.2.1.1. Accessible marking

The set of markings is the set M_i which can be reached by firing an S from the initial marking M_0 . $*M_0 = \{M_i \text{ such that } M_i[s \rightarrow M_j]\}$. As represented by the following example:

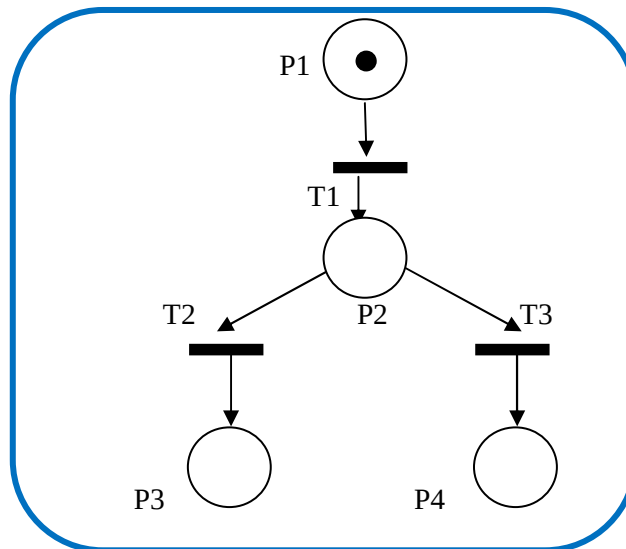


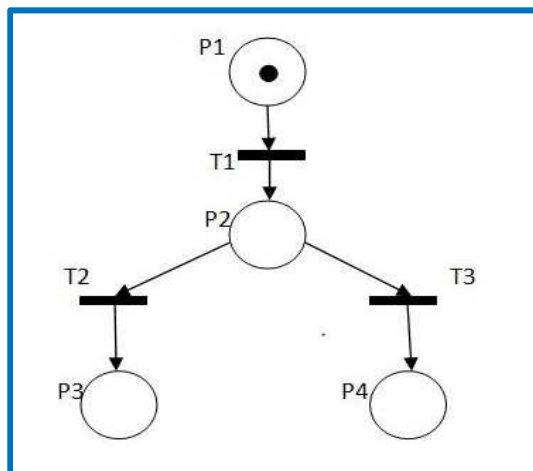
Figure 05: Marking set.

* $M0=\{M0\ M1\ M2\ M3\}$ avec $M0=[1\ 0\ 0\ 0]$; $M1=[0\ 1\ 0\ 0]$; $M2=[0\ 0\ 1\ 0]$; $M3=[0\ 0\ 0\ 1]$.

1.2.1.2 Accessible marking graph (reachability graph)

We use the marking graph when the number of accessible markings is finite.

Example: in this example we show the accessible marking and its corresponding graph.



The corresponding marking graph:

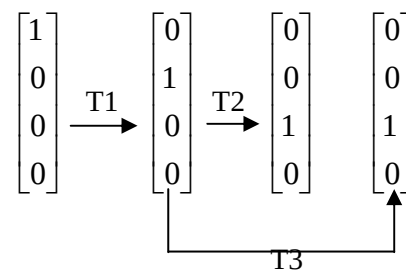


Figure 06: The accessible marking graph.

1.2.2. Firing a transition

We can say that an enabled transition can fire. If all the entry places of the transition contain at least one token. Firing consists of removing a token from each of the entry places and adding a token to each of the exit places of the same transition. Below we have a concrete example of firing a transition.

Example: firing a source transition consists of adding a token to each of these exit places as shown in this figure:

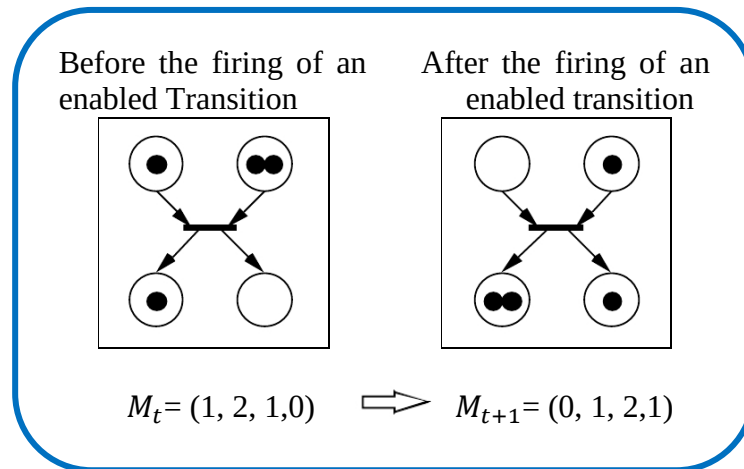


Figure 07: Firing a transition.

The general rules for the temporal evolution of a Petri net

◆ The general rules for evaluating simple labeled Petri nets are as follows:

- A.** Transition is enabled or validated when each of the places upstream has at least one token.
- B.** The network can only evolve by firing a single transition at a time, transition chosen from all those which are validated at this moment.
- C.** Firing a transition is indivisible and of zero duration.

If a transition is validated, this does not imply that it will be immediately taken. These rules indeed introduce a certain indeterminism into the evolution of Petri nets, since they can pass through different states whose appearance is conditioned by the choice of the transitions that can fire. This operation represents real situations quite well where there is no priority in the succession of events. **[Kaiser01]**

1.2.3. Firing a sequence

In a Petri net, firing a sequence S is a sequence of transitions $T_i, T_j, \dots, T_k$ which can be fired successively from a given marking. Only one transition can be completed at a time.

We note: $M_i[S \rightarrow M_j]$ or $M_i[S \rightarrow M_j]$ from the marking M_i , firing the sequence S results in the marking M_j . Example: $T_1 T_2$ and $T_1 T_3$ are two sequence of firings. **[RDP]**

$M_0 [T_1 T_2 \rightarrow M_1]$ et $M_0 [T_1 T_3 \rightarrow M_2]$ avec $M_1 = [0 \ 0 \ 1 \ 0]$ et $M_2 = [0 \ 0 \ 0 \ 1]$.

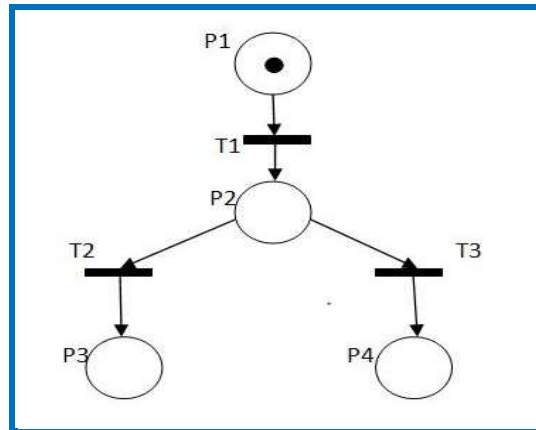


Figure 08: Firing of a sequence.

2. Autonomous and non-autonomous PNs

- 2.1. Autonomous PNs:** describes the operation of a system which evolves autonomously, i.e. whose firing times are not known, or not indicated. (Example: The cycle of the seasons. The moment of transition from summer to autumn is unknown.).

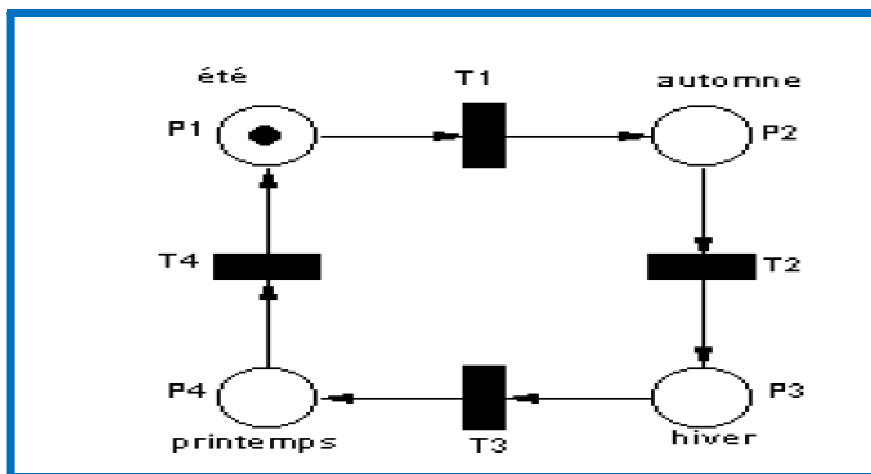


Figure 09: Autonomous PN

- 2.2. Non-autonomous PNs:** describes the operation of a system whose evolution is conditioned by external events or by time (Example: Starting an engine). A non-autonomous PN is Synchronized and/or Timed.

3. Particular Petri nets

State graph: An unmarked Petri net is a state graph if and only if every transition has exactly one entry place and one exit place. Example: transitions T1, T2, T3, T4 and T5 have one entry place and one exit place. **[Kurt03]** As detailed in this figure:

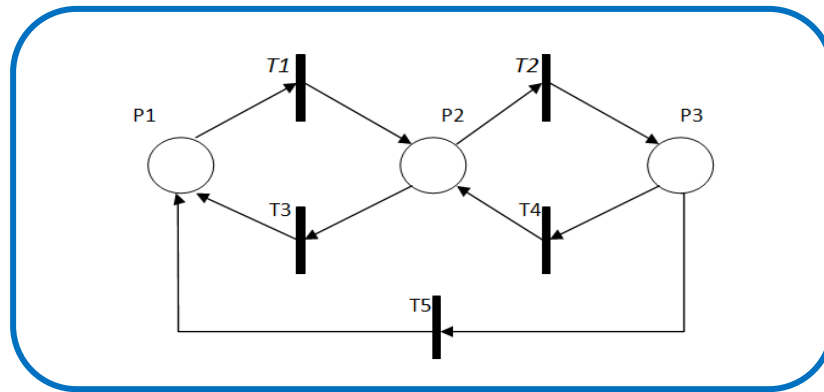


Figure 10: State graph.

- 3.1. **Event graph:** A Petri net is an event graph if and only if each place has exactly one input transition and one output transition as shown below. [Kurt03]

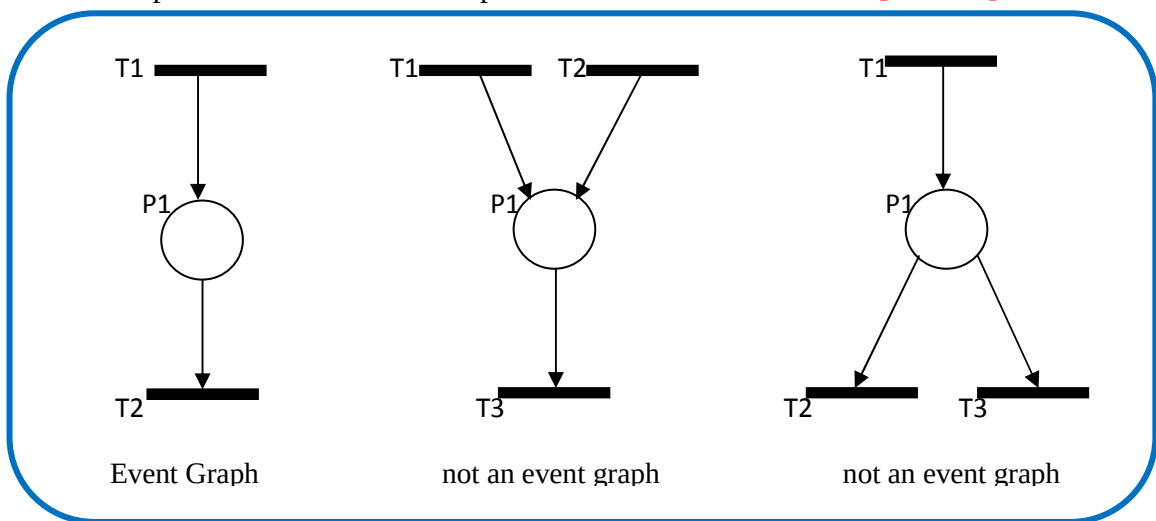


Figure 11: Event Graph.

- 3.2. **Conflict-free PN:** Say that a PN is a conflict-free network if and only if each place has at most one exit transition. An RDP with conflict is a network which therefore has a place with at least two output transitions. A conflict is noted: $[P_i, \{T_1, T_2, \dots, T_n\}]$; with $T_1, T_2, \dots, T_n$ being the output transitions of place P_i . As shown in figure 12 [Kurt03]

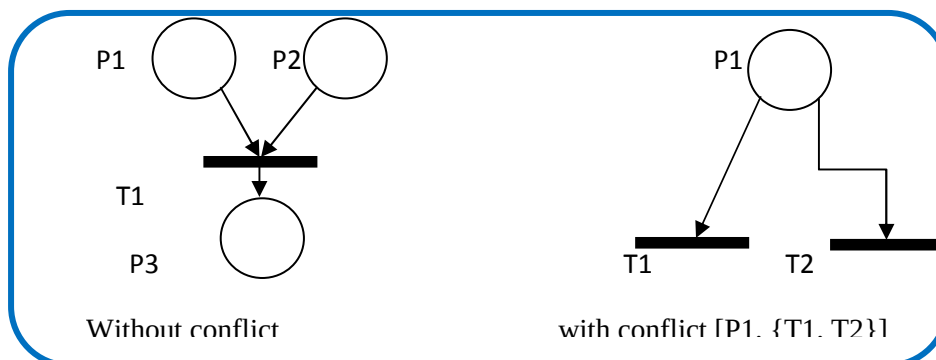


Figure 12: PN With and without conflict.

- 3.3. Free choice RDP:** A free-choice RDP is a network in which for any conflict $[P_i, \{T_1, T_2, \dots, T_n\}]$ none of the transitions $T_1, T_2, \dots, T_n$ has no other input place than P_i . As expressed in figure 13. [Kurt03]

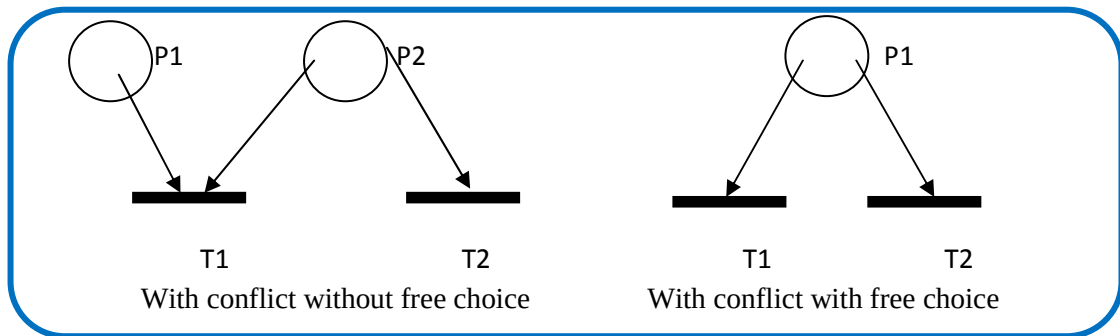


Figure 13: PN With and without free choice

- 3.4. Simple PN:** A simple Petri net is a PN in which each transition can only be affected by at most one conflict. As shown in Figure 14. [Kurt03]

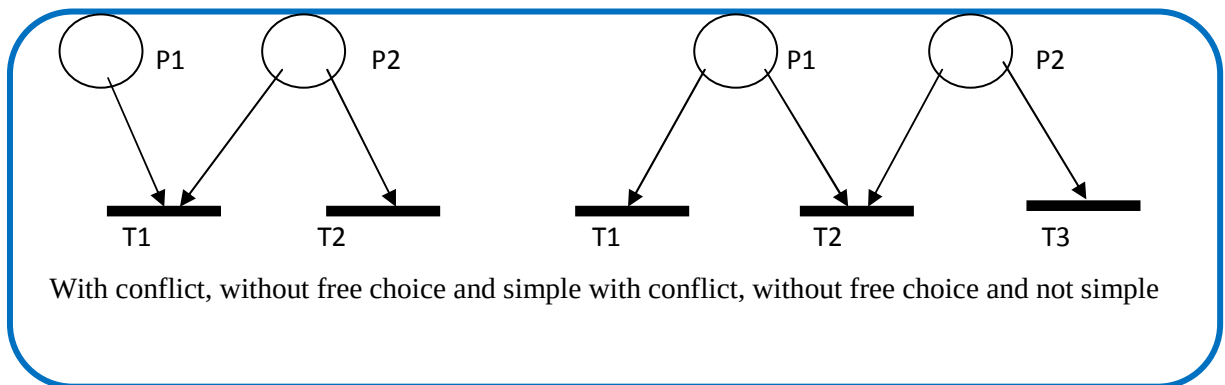


Figure 14: Simple PN.

- 3.5. Pure PN:** A pure PN is a network in which there is no transition having an entry place which is at the same time an exit place of this transition. [Kurt03]

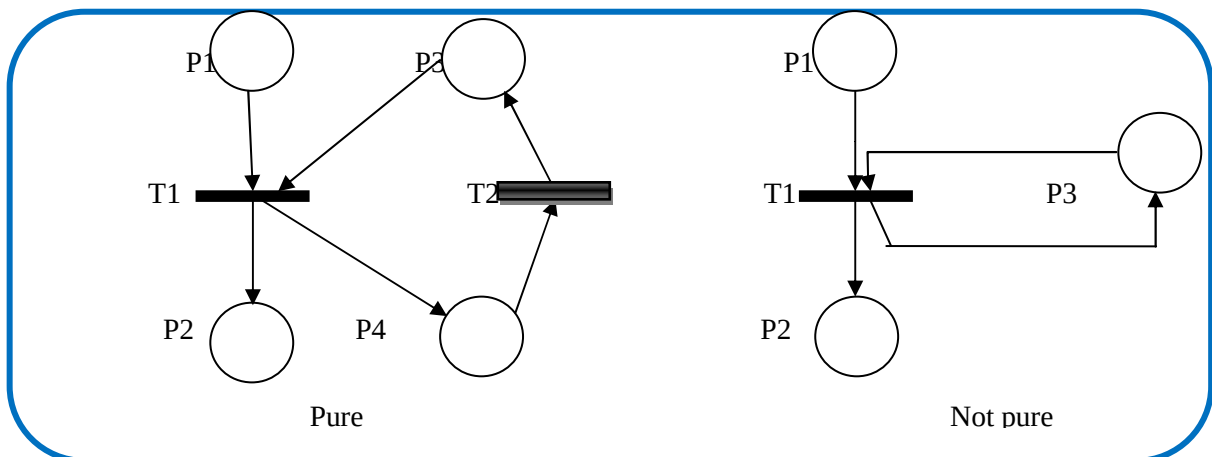


Figure 15: Pure PN.

- 3.6. Generalized PN:** A generalized PN is a PN in which weights (numbers strictly positive integers) are associated with arcs. If an arc (P_i, T_j) has a weight K : the transition T_j is only firing if the place P_i has at least K tokens. Firing consists of removing K tokens from place P_i . If an arc (T_j, P_i) has a weight K : firing the transition adds K tokens to place P_i . When the weight is not indicated, it is equal to one by default. [Kurt03] As it is expressed in the following figure:

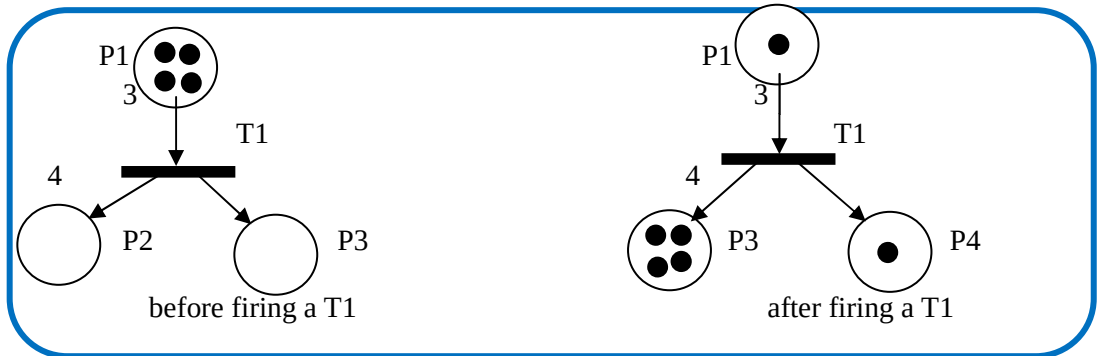


Figure 16: Generalized PN

- 3.7. Capacitated PN:** A capacity PN is a PN in which capacities (strictly positive integers) are associated with places. Firing an entry transition of a place P_i whose capacity is $\text{cap}(P_i)$ is only possible if the firing does not lead to a number of tokens in P_i which is greater than $\text{cap}(P_i)$. [Kurt03]

Example: firing T_i leads to 3 tokens in P_2 from which T_1 can no longer be fired.

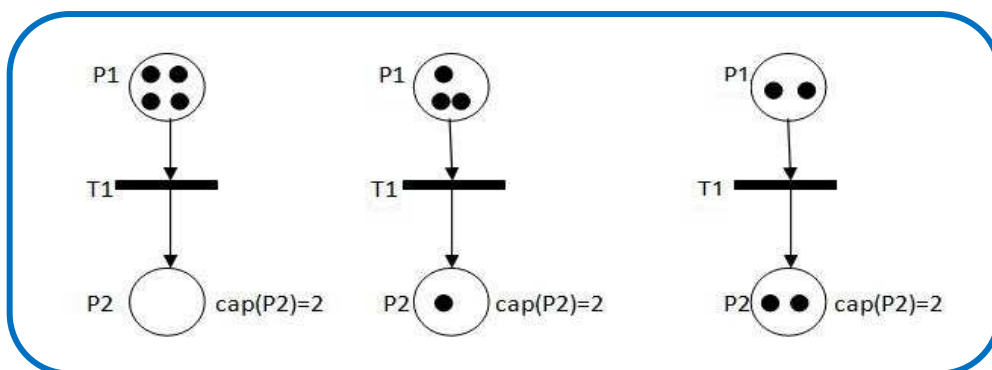


Figure 17: Capacitated PN.

- 3.8. PN with priorities:** In such a network, if we reach a marking such that several transitions are enabled, we must fire the transition which has the highest priority. In the following example we presented a priority PN. As shown schematically in the following figure. [Kurt03]

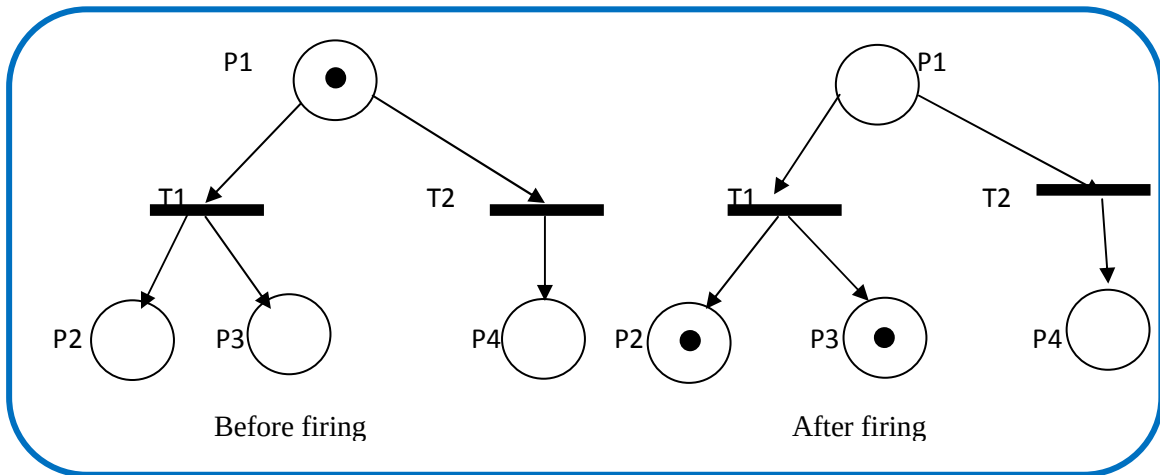


Figure 18: PN with priority.

- 3.9. PN without loop:** A PN without loop is such that there exists a transition T_j and a place P_i which is both entry place and exit place of T_j , then T_j has at least one other entry place.

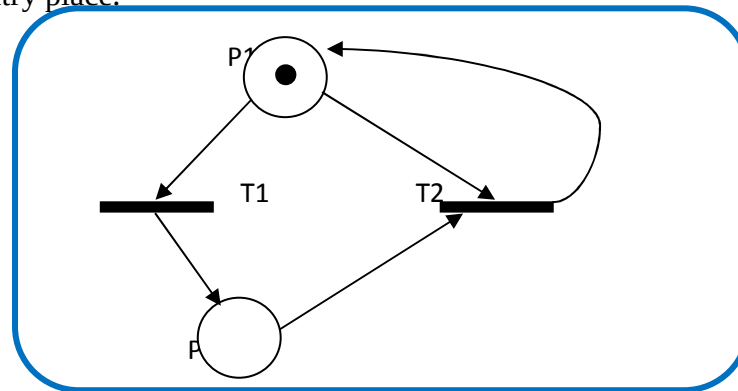


Figure 19: PN without loop.

- 3.10. PN with inhibitor arcs:** An inhibitor arc is a directed arc that starts from a place to end in a transition (and not the other way around). Its end is marked by a small circle. The presence of an inhibitor arc between a place P_i and a transition T_j means that the transition T_j is only validated if the place P_i does not contain any token. Firing the transition T_j consists of removing a token in each place located upstream of the transition with the exception of place P_i , and adding a token in each place located downstream of the transition.

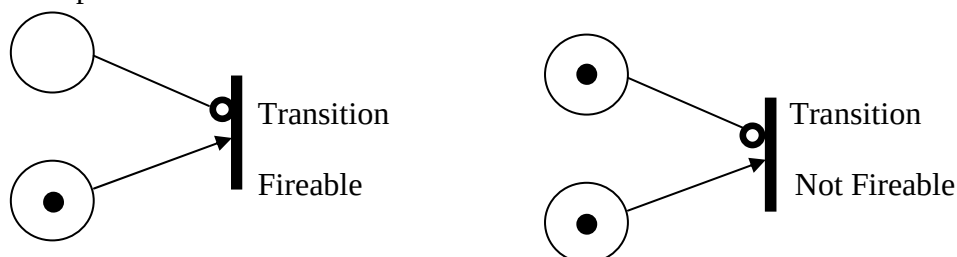


Figure 20: PN with inhibitor arcs

3.12. Continuous PN: Their particularity is that the marking of a place is a real number (positive) and no longer an integer.

4. The different types of Petri nets

There are several types of Petri nets, we cite in particular:

4.1. Timed Petri nets

An ordinary Petri net describes a causal relationship between events. An event “a” is the cause of “b”, an always precedes “b”, “a” and “b” are ordered in time. Time is taken into account qualitatively. Approaches will be presented if they allow time to be taken into account quantitatively. Time is directly associated with the Petri net, it is part of the controls instead of being rejected in the given part in an unstructured way. [Valette02]

◆ *Time associated with a place.*

For a place representing an activity, it is simply a matter of noting the duration of this activity. We can consider that, in fact, the place concerned bursts into a “place-transition-place” sequence. The first place corresponds to the current activity, the transition corresponds to the elapsed time event and the last place corresponds to a possible wait (synchronization with other activities) after the end of the activity (figure 2.21). While the activity is in progress the token cannot be used to fire a transition. It is said to be unavailable. Once the activity is completed, the token becomes available and transition t2 can possibly be taken.

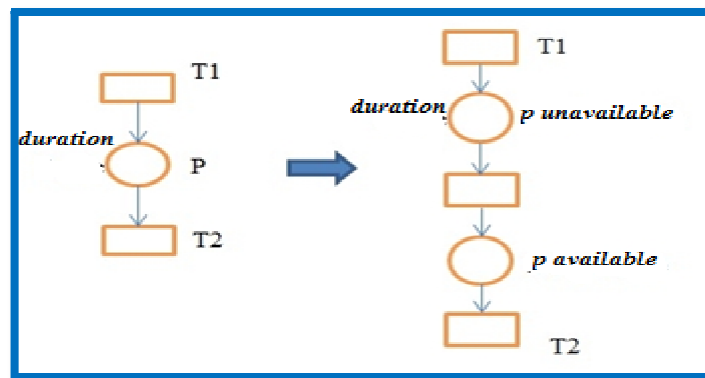


Figure 21: Time Associated with a place.

◆ *Time associated with a transition*

This association only makes sense if the transition is interpreted as an (interruptible) activity and not as an instantaneous event. This transition is broken down into a “transition-place-transition” sequence. The first transition corresponds to the instantaneous event of start of activity (we remove all the tokens), the place is used to memorize the current activity and the last transition corresponds to the instantaneous event of end of activity (puts the tokens in the exit

places). Firing the first transition following place p_i of the exploded form corresponds to the reservation of tokens (the reserved tokens can no longer be used to fire a transition other than T). After firing the second transition the tokens are released.

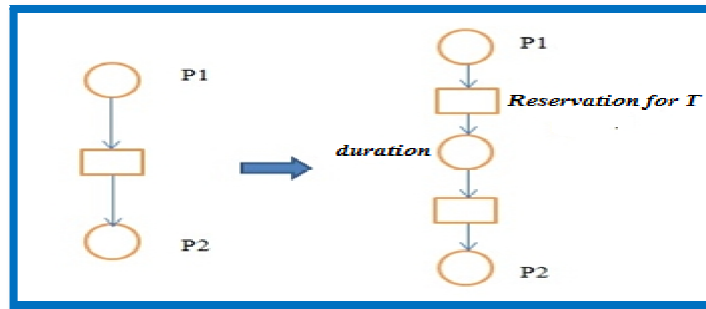


Figure 22: Time associated with a transition.

4.2. Stochastic Petri nets

Stochastic Petri nets were introduced by Natkin [Natkin80] and Molloy [Molloy81] in order to respond to certain problems of quantitative evolution of industrial computer systems. In stochastic Petri nets, the delays associated with transitions are random unlike the deterministic and constant durations associated with timed PNs. These times are modeled by random variables whose most common law is the exponential law which allows the marking graph to be approximated to a homogeneous Markovian process. Stochastic Petri nets are widely used in operational safety.

4.3. Colored Petri nets

Colored Petri nets make it easier to model large systems. They are of great interest for modeling certain complex systems. The principle consists of representing the information by the place/marked sets. The marks of each place are associated with a color (or identifier). Firing these marks can be done in several ways depending on the colors associated with the transitions. The relationship between the firing colors and the colored markings is defined by functions associated with the arcs. This type of Petri nets interests us in our work and the following part shows the Colored PNs in explicit way.

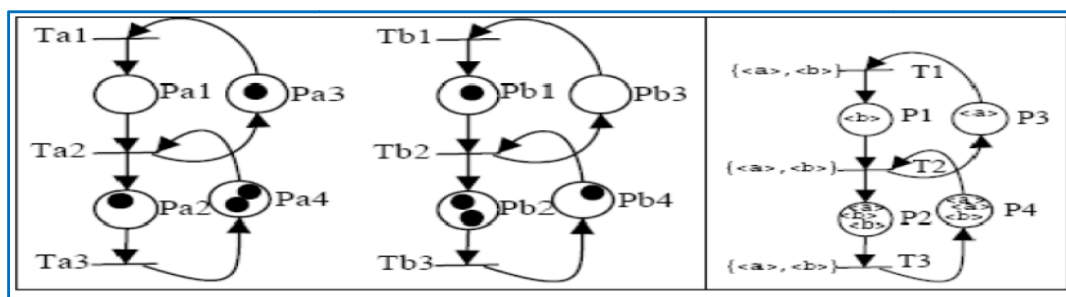


Figure 23: PN (Left) and colored PN (right).

5. Matrix representation:

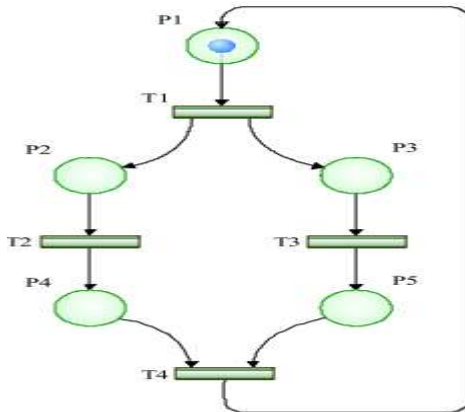
A matrix representation of a PN is used to simplify the analysis and verification tasks carried out on a PN model. Acting on a graphical representation of a PN model is a delicate task by comparing it with a matrix representation. It is possible to represent the function W (weight function) by matrices. [ElMansouri; 2009].

A- Post-incidence matrix: contains the number of trademarks registered in p_i when firing the transition t_j . It is an $n \times m$ matrix such that $\text{post}(i,j) = W(t_j, p_i)$ with n the number of places and m the number of transitions in the PN.

Post =

$$\begin{matrix} & T1 & T2 & \dots & Tm \\ \begin{matrix} P1 \\ P2 \\ \vdots \\ Pn \end{matrix} & \begin{pmatrix} \\ \\ \\ \end{pmatrix} \end{matrix}$$

Each element of this matrix $\text{Post}(P_i, T_j)$ corresponds to the number of tokens to add to P_i in firing T_j . Example: the Post-incidence matrix is:



$$\begin{matrix} & T1 & T2 & T3 & T4 \\ \begin{matrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

B- Pre-incidence Matrix: It is the matrix $m \times n$ with coefficients in $\mathbb{N}$ such that $\text{pre}(i,j) = W(p_i, t_j)$ indicates the number of marks that the place p_i must contain for the transition t_j to become firable. n the number of places and m the number of transitions in the PN.

Pre = Each element of this matrix $\text{Pre}(P_i, T_j)$ corresponds to the number of tokens to remove in P_i in firing T_j . Example: the Pre-incidence matrix is

$$\begin{matrix} & T1 & T2 & \dots & Tm \\ \begin{matrix} P1 \\ P2 \\ \vdots \\ Pn \end{matrix} & \begin{pmatrix} \\ \\ \\ \end{pmatrix} \end{matrix}$$

$$\begin{array}{c}
 \begin{array}{ccccc}
 & T1 & T2 & T3 & T4 \\
 \begin{pmatrix}
 p1 & 1 & 0 & 0 & 0 \\
 p2 & 0 & 1 & 0 & 0 \\
 p3 & 0 & 0 & 1 & 0 \\
 p4 & 0 & 0 & 0 & 1 \\
 p5 & 0 & 0 & 0 & 1
 \end{pmatrix}
 \end{array}
 \end{array}$$

C-Incidence Matrix : It is a matrix with n rows and m columns with n the number of places and m the number of transitions in the PN. It can be calculated as follows:

$$C = \text{Post} - \text{Pre} =$$

$$\begin{array}{c}
 \begin{array}{c}
 P1 \\
 P2 \\
 \vdots \\
 Pn
 \end{array}
 \begin{array}{c}
 \begin{pmatrix}
 T1 & T2 & \dots & Tm
 \end{pmatrix}
 \end{array}
 \end{array}$$

Each element of this matrix $C(P_i, T_j)$ corresponds to the number of tokens to add minus the number to remove in P_i by firing.

The marking of a Petri net is represented by a vector of dimension m with coefficients in $\mathbb{N}$. The rule for firing a Petri net is defined by:

$$M'(p) = M(p) + C(p, t).$$

6. Properties of Petri nets

a. Boundedness PN:

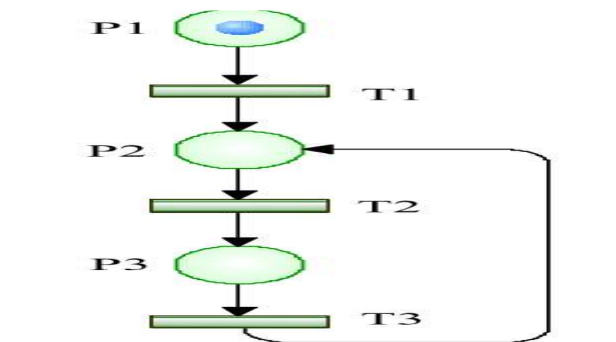
- **A place P_i** is said to be bounded for an initial marking M_0 if for any marking accessible from M_0 the number of tokens in P_i is finite.
 - a. **A PN** is bounded for an initial marking M_0 if all places are bounded for M_0 . If for any marking M apartment to the set of markings accessible from M_0 (denoted $*M_0$), we have $M(P_i) \leq K$ where K is a natural integer, we say that P_i is K -bounded. If the property is true for any place we say that this PN is **K-bounded**.

b. Liveness:

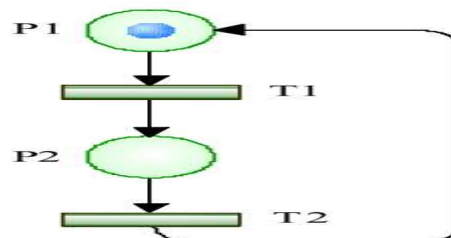
For an initial marking M_0 if for any accessible marking $M_i \in *M_0$, there exists a Firing sequence S which contains T_j starting from M_i (i.e. there will always remain a possibility of firing T_j). The concept of liveness is closely related to the complete absence of deadlocks in operating systems.

A Petri net (N_0, M_0) is said to be live (or equivalently M_0 is said to be a live marking for N) if, no matter what marking has been reached from M_0 , it is possible to ultimately fire any transition of the net by progressing through some further firing sequence. This means that a live Petri net guarantees deadlock-free operation, no matter what firing sequence is chosen.

Example: In the following example, the transitions T_2 and T_3 are alive while T_1 is not! In fact, it can only be fired at the start.



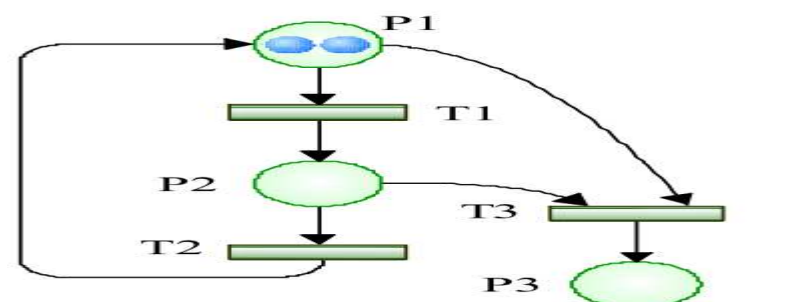
- ◆ A Petri net is said to be live if for all transitions there is a way to fire transition in any marking 'M' reachable from the initial marking M_0 i.e. ' $\forall M' \in R(M_0)$ and $\forall t \in T$, $\exists M \in R(M')$ such that t is enabled in M '.



- ◆ **Blocking:** A blocking (or sink state) is a marking such that no transition is validated.

A marking 'M' reachable from the initial marking M_0 is a deadlock if none of the transitions of the Petri net is enable in 'M'.

- A PN is said to be without blocking for an initial marking M_0 if $\forall$ accessible marking $M_i \in *M_0$, it is without blocking.



Interpretation:

Liveness indicates that the system represented is deadlock-free, but also that there is no dead branch in the graphical model and therefore no incomplete specification.

7. Petri net analysis methods

The modeling of a system must allow the analysis of its properties. Petri nets offer powerful analysis techniques for validating behavioral models of discrete event systems. Among these techniques, we cite the marking graph, the matrix equation and the reduction of petri nets. [Behri ; 2010].

- **The marking graph:** This involves constructing the graph of all the markings in the network. The properties are then deduced using graph theory techniques.
- **The matrix equation:** This method consists of finding a matrix representation of the network, linear algebra techniques then make it possible to obtain the properties of the network.
- **Reduction of PNs:** For the analysis of the properties of a PN of significant size, the use of the marking graph or the matrix equation proves insufficient. The objective of the reduction technique is to present rules making it possible to obtain from a marked PN, a simpler marked PN, with a reduced number of places and transitions. The power of Petri nets lies in the existence of a battery of analysis tools such as:

- INA (*Integrated Net Analyzer*) [INA]
- PEP (*Programming Environment based on Pétri nets*) [Pep]
- TINA [*Tina*], etc. ...