Petri Nets

Etienne Caruana

Petri Nets (PN) provide a graphical approach to the modelling of communicating systems. PN have been introduced by *Carl A. Petri* in his dissertation presented in 1962. The purpose was to analyse communication systems. Further study on this subject has led to a vast applicability of PN in many sectors like in I.T. and manufacturing.

The reason for such a vast applicability of PN is due to the fact that PN hold a considerable **modelling power** as well as efficient methods for proper **performance analysis** of the system under study. In fact, PN may be shown to be effective in the modelling of concurrency, conflict and synchronisation.

Description of a PN: A PN is a bipartite, directed graph. The set N of nodes is divided into the set of **transitions** T and **places** P, i.e. $N = P \cup T$, $P \cap T = \emptyset$. Elements of $T \times P$ and $P \times T$ are said to be the **arcs** of the net. Places contain **tokens**. Tokens are usually represented by dots inside the places or by a number indicating the number of tokens that reside inside each place For

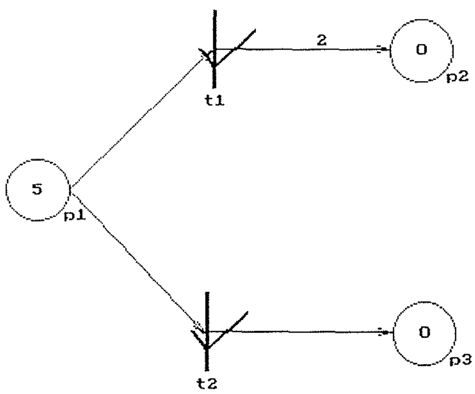

Figure 2: A simple Petri Net

example, in Figure 2:

- $P = \{p_1, p_2, p_3\}$
- $T = \{t_1, t_2\}$
- Set of input arcs $P \times T = \{(p_1, t_1), (p_1, t_2)\}$

- Set of output arcs $T \times P = \{(t_1, p_2), (t_2, p_3)\}$

Initially, this net has 5 tokens inside p_1 and 0 tokens inside p_2 and p_3.

Place p_1 is said to be an **input place** of transitions t_1 and t_2, while places p_2 and p_3 are said to be **output places** of t_1 and t_2 respectively.

Given a transition (place) t (p), then the set of output places (transitions) is denoted by $p\cdot$ ($t\cdot$), while the set of input places (transitions) is denoted by $\cdot p$ ($\cdot t$). Therefore, from Figure 1:

- $p_1\cdot = \{t_1, t_2\}$; $t_1\cdot = \{p_2\}$; $t_2\cdot = \{p_3\}$
- $\cdot t_1 = \{p_1\}$; $\cdot t_2 = \{p_1\}$; $\cdot p_2 = \{t_1\}$; $\cdot p_3 = \{t_2\}$

Tokens flow from one place to another by the firing of **transitions**. The firing of each transition depends both on the number of tokens inside each place and the **multiplicity** of each arc. The multiplicity of an arc is marked by a number attached to the arc. If no number is attached to an arc, the multiplicity of the arc is taken to be one.

A transition may fire only if the number of tokens inside each of its input places is greater than or equal to the multiplicity of the corresponding arc. If the transition is in a condition to be fired, then it is said to be **enabled**. For example, in Figure 2, both t_1 and t_2 are enabled since the number of tokens inside p_1 is 5 and the multiplicity of input arcs is 1 for both.

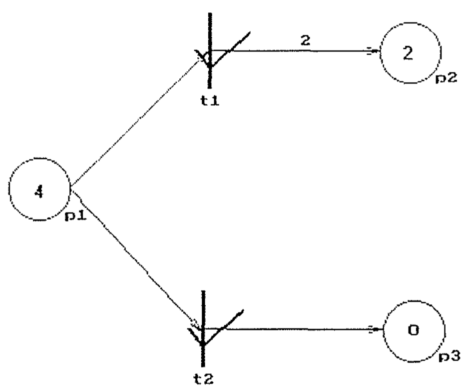

Figure 3: The Petri Net after firing t_1

When a transition fires, it removes a number of tokens from each input place equivalent to the multiplicity of its input arcs and adds a number of tokens to its output places, equivalent to the multiplicity of its output arc. Figure 3 represents the marking of the PN after firing t_1.

If we fire t_1 4 times more, the marking of p_1 will eventually decrease to 0. Then, t_1 and t_2 will not be able to fire. In that case, we say that the net is in a state of **deadlock** - a state in which no flow of tokens inside the net is possible.

Applications of PN: The following is a list of attributes of PN which make them useful for modelling:

1. Petri Nets capture the precedence relations and structural interactions of stochastic, concurrent, and asynchronous events. In addition, their graphical nature helps to visualise such complex systems.

2. Conflicts and buffer sizes can be modelled easily and efficiently.

3. Deadlocks in the system can be detected.

4. Petri net models represent a hierarchical modelling tool with a well-developed mathematical and practical foundation.

5. Various extensions of PN, such as timed PN, stochastic (timed) PN, coloured PN, and predicate/transition nets, allow for both qualitative and quantitative analyses of resource utilisation, effect of failures, and throughput rate, to name a few.

6. Petri net models give a structured framework for carrying out a systematic analysis of complex systems. Various software packages have been developed for this purpose.

In general, for modelling purposes, places represent **resources** such as machines or parts of a buffer. The meaning of a token inside a place is generally deduced according to the definition of the place. For example, a token in a place representing a state of a machine would represent the availability of the machine to do the required job while a token inside a place representing a store of a manufacturing system would represent the number of items produced by the system.

In general, a transition firing represents an activity, which begins and ends by two consecutive events (represented by places). For example, a machine representing a transition labelled 'machine repair' should have an input place representing 'machine damage' and have an output place representing 'machine working'.

The following example illustrates a Petri-Net model of a simple communication system.

Figure 4 depicts a communication system made of three robots R1, R2, R3 and two conveyors C1 and C2. Each conveyor operates on the two robots next to

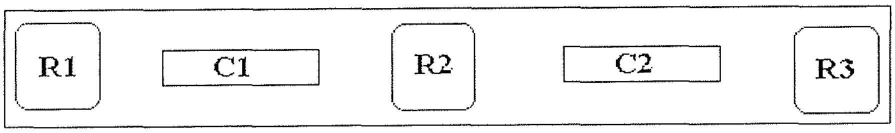

Figure 4: Three robots working with two conveyors

it. Both conveyors first use the robot on the left-hand side and then the robot on the right-hand side. No conveyor may be used by two robots at the same time, implying that R2 needs to be shared between C1 and C2. It is assumed that both conveyors remove their robots synchronously after their operations.

Tables 1 and 2 respectively give the places and the transitions which are needed to represent this system together with their description in this context.

Place	Meaning
p_1	C1 needs R1
p_2	C1 needs R2
p_3	C1 has both R1 and R2
p_4	C2 needs R2
p_5	C2 needs R3
p_6	C2 has both R2 and R3
p_7	R1 free
p_8	R2 free
p_9	R3 free

Table 1: Introducing Places

Transition	Meaning
t_1	C1 takes R1
t_2	C1 takes R2
t_3	C1 removes both R1 and R2
t_4	C2 takes R2
t_5	C2 takes R3
t_6	C2 removes both R2 and R3

Table 2: Introducing Transitions

The corresponding Petri Net would be the one shown in Figure 5. The initial marking of this net is $m_0(p_1) = m_0(p_4) = m_0(p_7) = m_0(p_8) = m_0(p_9) = 1$, with all the other places having a null marking. This marking is representing an initial situation where each robot is free and each conveyor is waiting for its left robot.

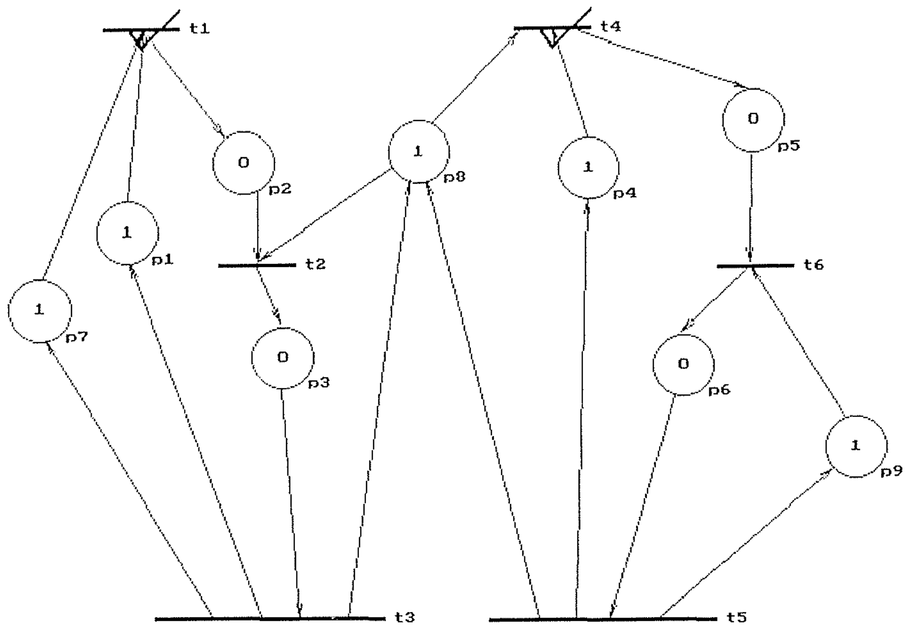

Figure 5: Petri Net of conveyor/robot problem

Note that initially the enabled transitions t_2 and t_4 are sharing the same input place, namely p_8.

Upon firing t_1, t_4 is disabled and vice versa. The sharing of place p_8 is representing a conflicting situation, where a decision has to be made of which transition to fire, analogous to deciding to either work with conveyor C1 or else with conveyor C2.