Discrete Event Simulation

- Models execution of PN's by establishing a series of firing events for the transitions of the system.
- To advance the system an event is chosen and executed.
- Each event execution may produce future events.

Lecture 5

Extensions to PN's and a formal definition of TTPN's and PTPN's.

Algorithm Outline

- For each probability distribution compute a sequence of delay values.
- For each transition which has all its inputs obtain the first delay value in the sequence and schedule an event to fire that transition.
- Determine any other enabled transitions and schedule future events for them.

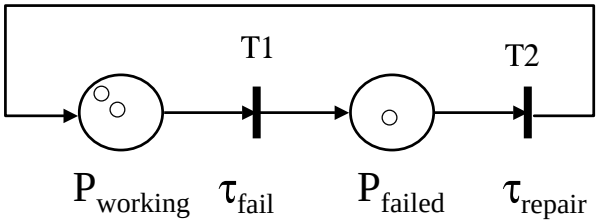
Executing Timed PN's

To model the execution of Timed petri-nets a transition becomes enabled if,

- the required tokens are available, **AND**
- the time delay at the place or transition has expired.

How is this achieved in practice?

Example

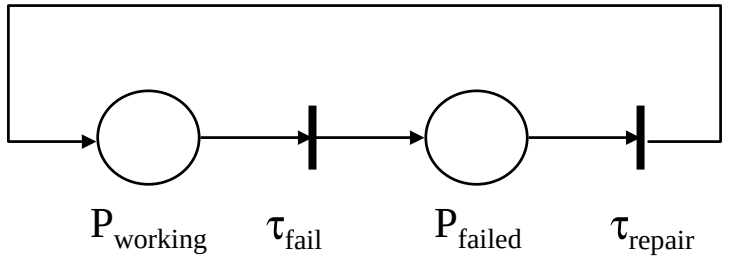


τ_{fail}	τ_{repair}
5	3
2	3
8	4
9	2
3	1

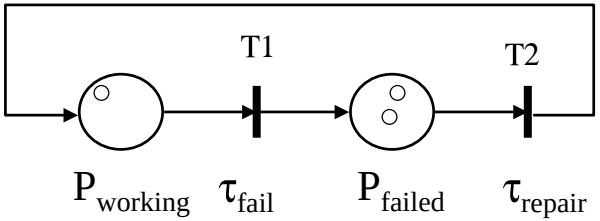
t	event
5	T1
7	T1
8	T2
15	T1

Example

- Modelling fail and repair of equipment.



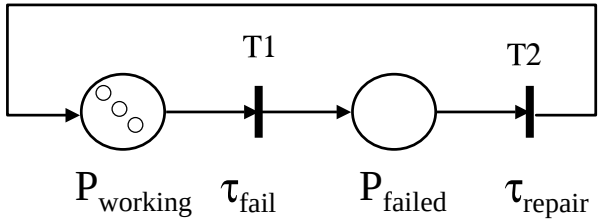
Example



τ_{fail}	τ_{repair}
5	3
2	3
8	4
9	2
3	1

t	event
5	T1
7	T1
8	T2
10	T2
15	T1

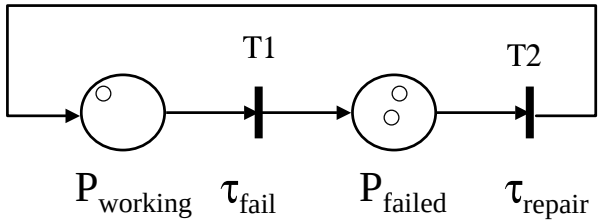
Example



τ_{fail}	τ_{repair}
5	3
2	3
8	4
9	2
3	1

t	event
5	T1
7	T1
15	T1

Example



τ_{fail}	τ_{repair}
5	3
2	3
8	4
9	2
3	1

t	event
5	T1
7	T1
8	T2
10	T2
15	T1

Notes

- Obviously this execution can continue for as long as needed.
- Using such a model one can determine
  - The average number of operating units.
  - The average interval between failures.
  - How often the number of operating units drops below a certain threshold.
  - etc.

Useful PN Structures

- There are standard PN structures which can be used to represent program activity.
- We have seen one, the model for mutual exclusion, and sequences.
- There are others for
  - fork/join
  - choice
  - alternation
  - producer/consumer