Petri Nets

small tutorial.

by *Simone Pellegrini*

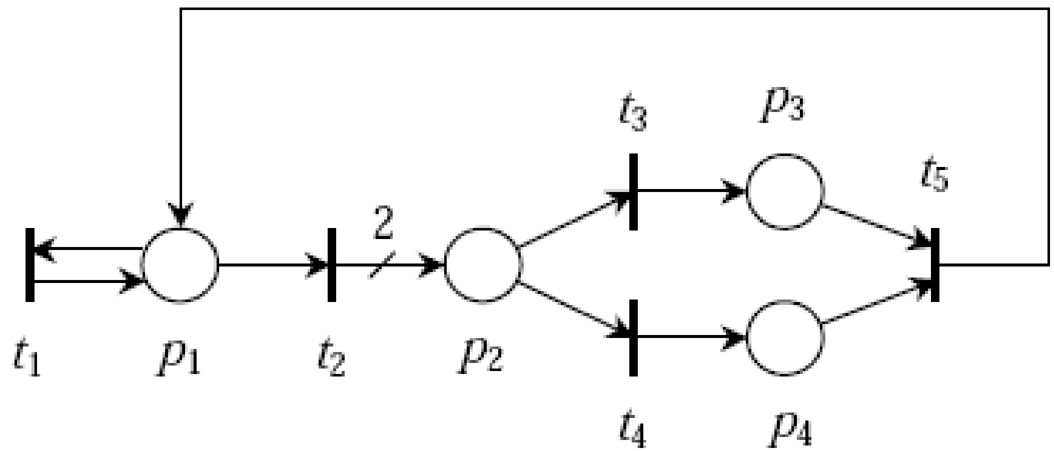
What is a Petri Net?

- Mathematical representations (*modeling language*) of *discrete distributed systems*;
- Invented in 1962 by *Carl Adam Petri* in his Ph.D thesis;
- It graphically depicts the structure of a *distributed system* as a directed bipartite graph;
- Execution of Petri nets is *nondeterministic* (well suited for modeling the *concurrent behavior* of distributed systems);

Definition

- A Place/Transition net is a structure *N = (P, T , Pre, Post)* where:
  - *P* is a set of ***places*** represented by circles, |P| = m;
  - *T* is a set of ***transitions*** represented by bars, |T| = n;
  - *Pre* : P × T → N is the pre-incidence function that specifies the arcs directed from places to transitions;
  - *Post* : T × P → N is the post-incidence function that specifies the arcs directed from transitions to places.

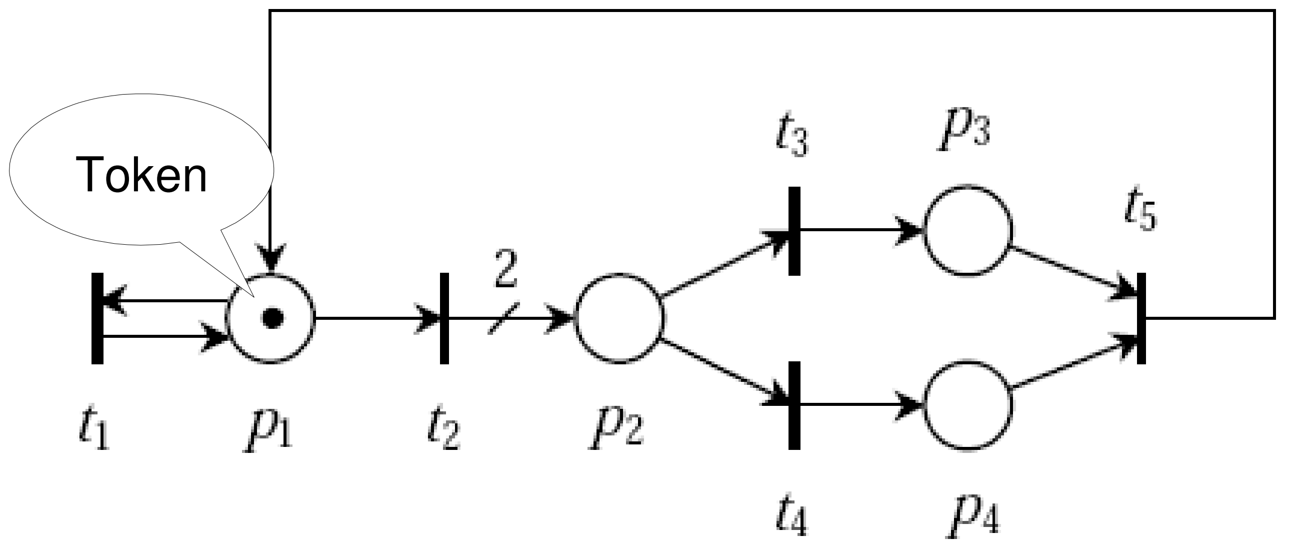
Petri Net



$$P = \{p_1, p_2, p_3, p_4\}, \quad T = \{t_1, t_2, t_3, t_4, t_5\}$$

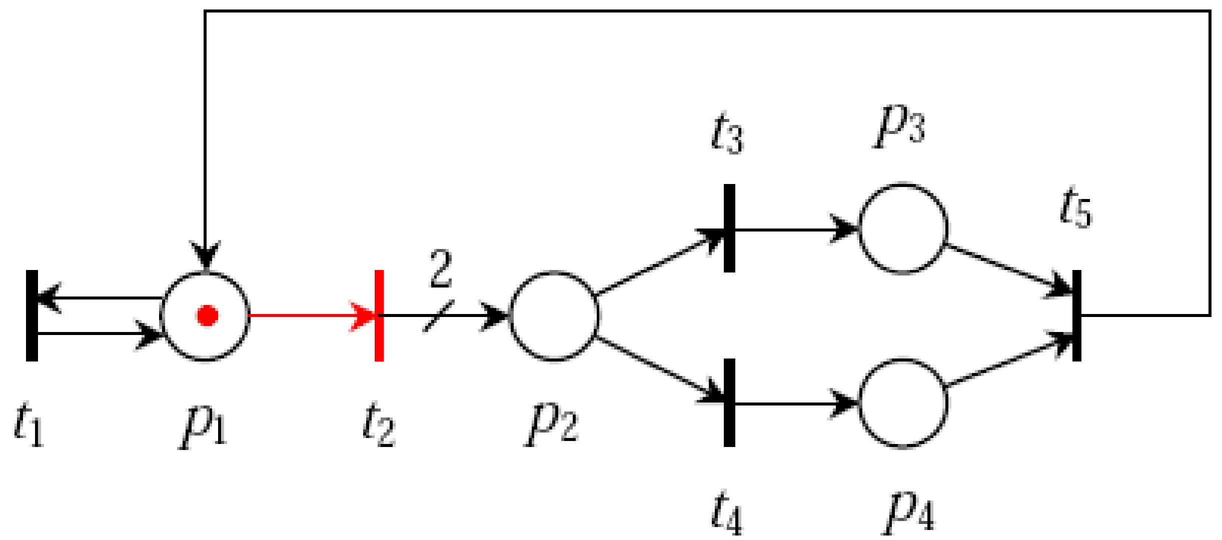
$$Pre = \begin{array}{cc} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \\ \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 \end{matrix} & \end{array} \qquad Post = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} \\ \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 \end{matrix} & \end{array}$$

Tokens and Marking



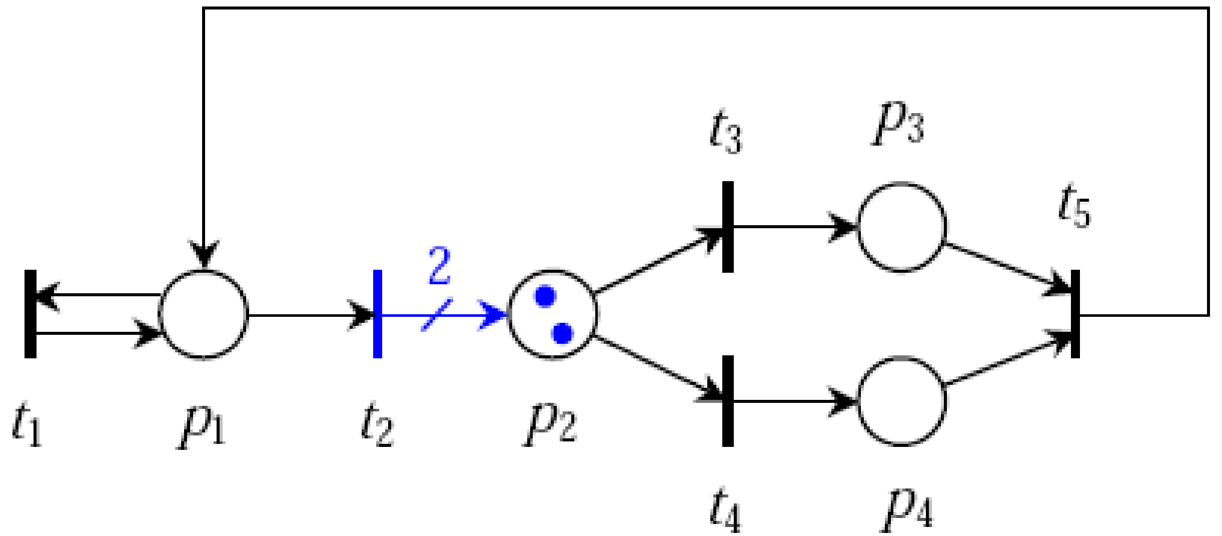
- A ***marking*** is a function *M : P → N* that associates to each place a non negative number of tokens. The initial marking is called M_0. (M_0 = $[1\ 0\ 0\ 0]^T$)

Enabling



- A transition is enabled if its each input place contains <u>at least as many tokens as the weight of the arc indicates</u>.
- An enabled transition can fire between time 0 and infinity.

Firing

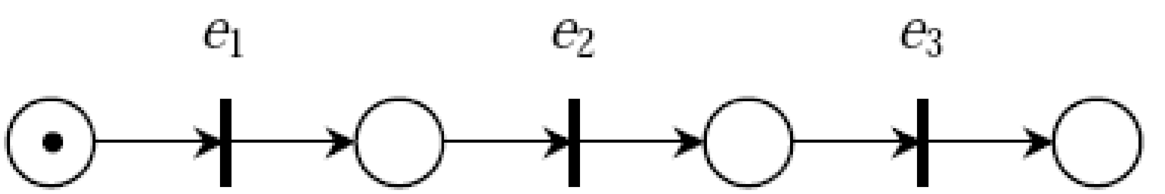


DEMO

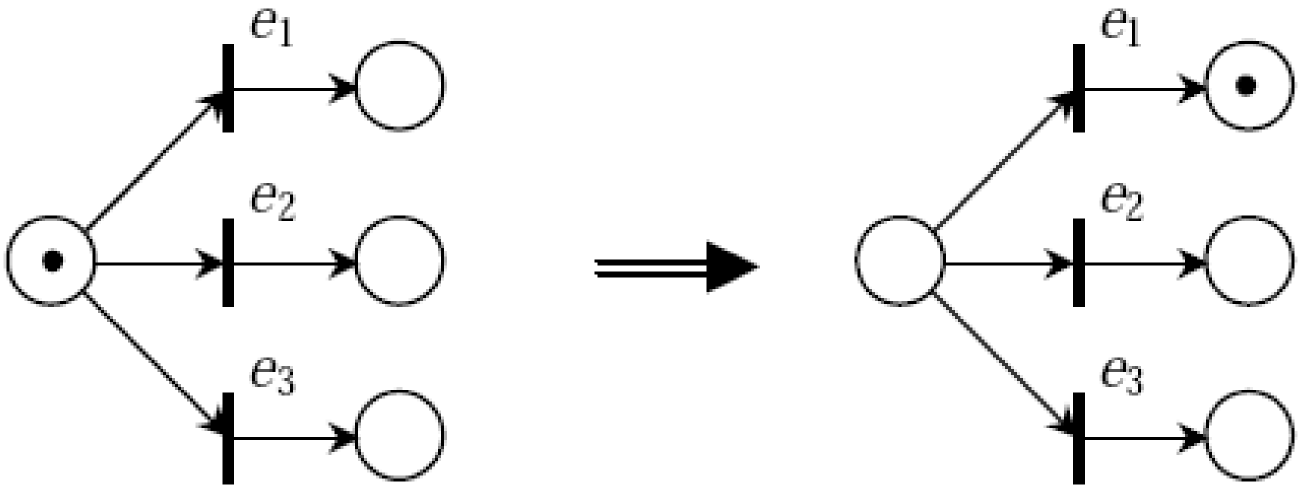
- Firing an enabled transition *transforms the state:*
  - subtract the input arc weights from the token counts of the input places;
  - add the output arc weights to the token counts of the output places.

Modeling structures (1/2)

- Sequence

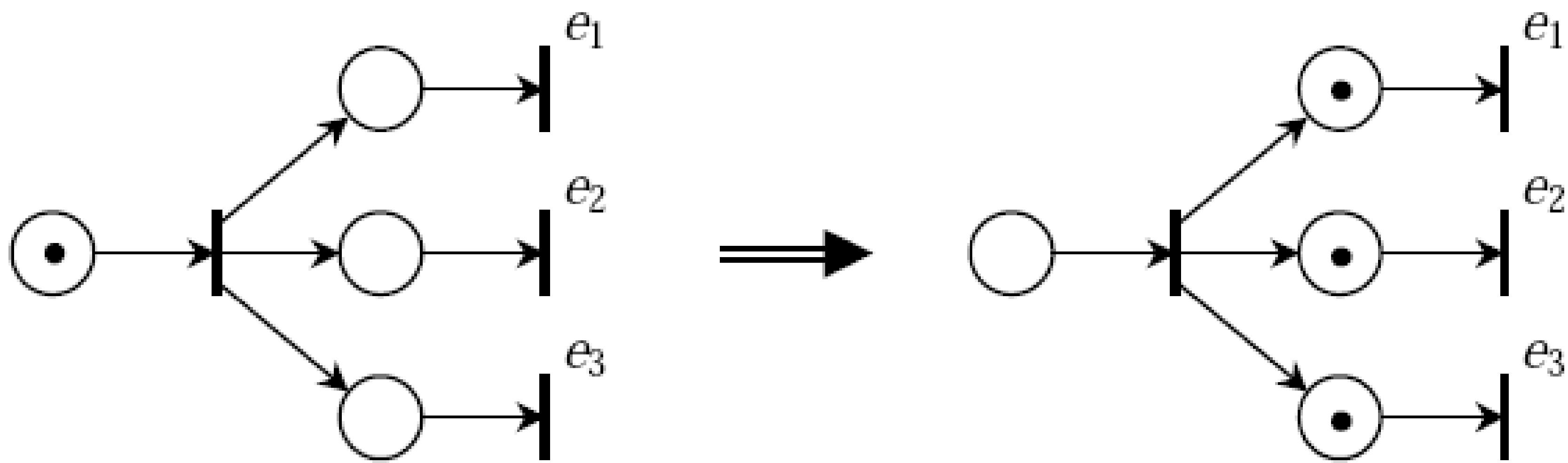


- Choice (non determinism)

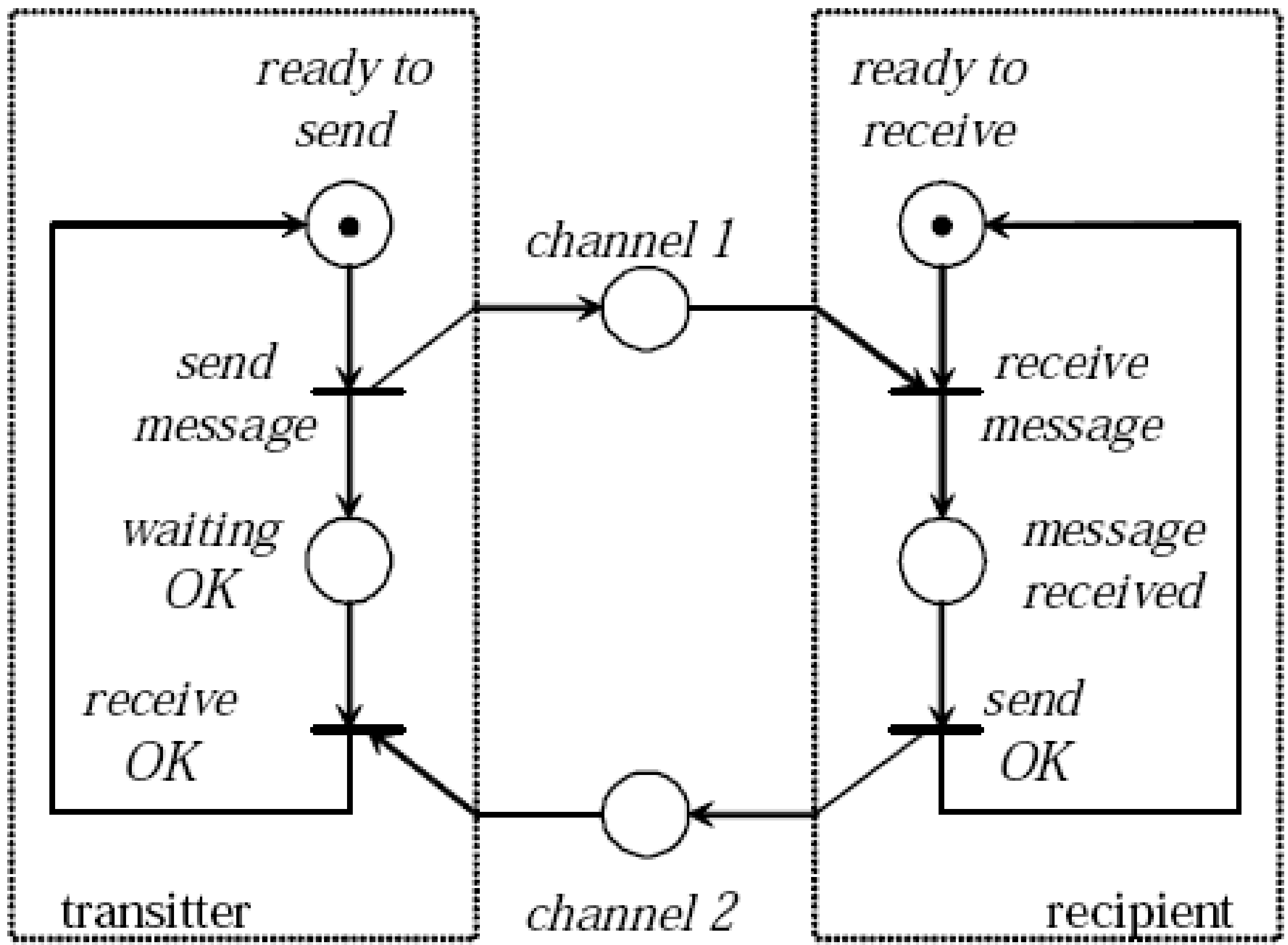


Modeling structures (2/2)

- Concurrency



The Producer/Consumer



DEMO

Petri Net Properties

- The ***state*** and the ***evolution*** of the net can be modeled in a formal way:
  - *state equation*,
  - *reachability* (*coverability*) *graph.*
- The *behavioral properties* depends on the structure of the net and the initial marking:
  - *reachability,*
  - *boundedness and safeness,*
  - *liveness.*

Reachability

- A sequence of transition firings can be seen as a sequence of markings:
  - **<u>DEF</u>:** *A marking M_k is reachable from initial marking M_0 if a sequence transition firings which transforms M_0 in M_k exists.*
- The reachability of the states can be represented with a *reachability graph*;
- It is used to check a wrong state such as an elevator moving while the door is open.

Boundedness and Safeness

- **DEF**: *A Petri net is k-bounded if the number of tokens in each place (for each possible evolution) of the net is equal or less than the integer positive number k*.
- A Petri net is *safe* if it is *1-bounded*.
- A Petri net is *bounded* if all its reachability graphs all have a finite number of states.
- Boundedness of a Petri net is used to model that system resources, such as CPUs, are bounded.

Liveness (1/2)

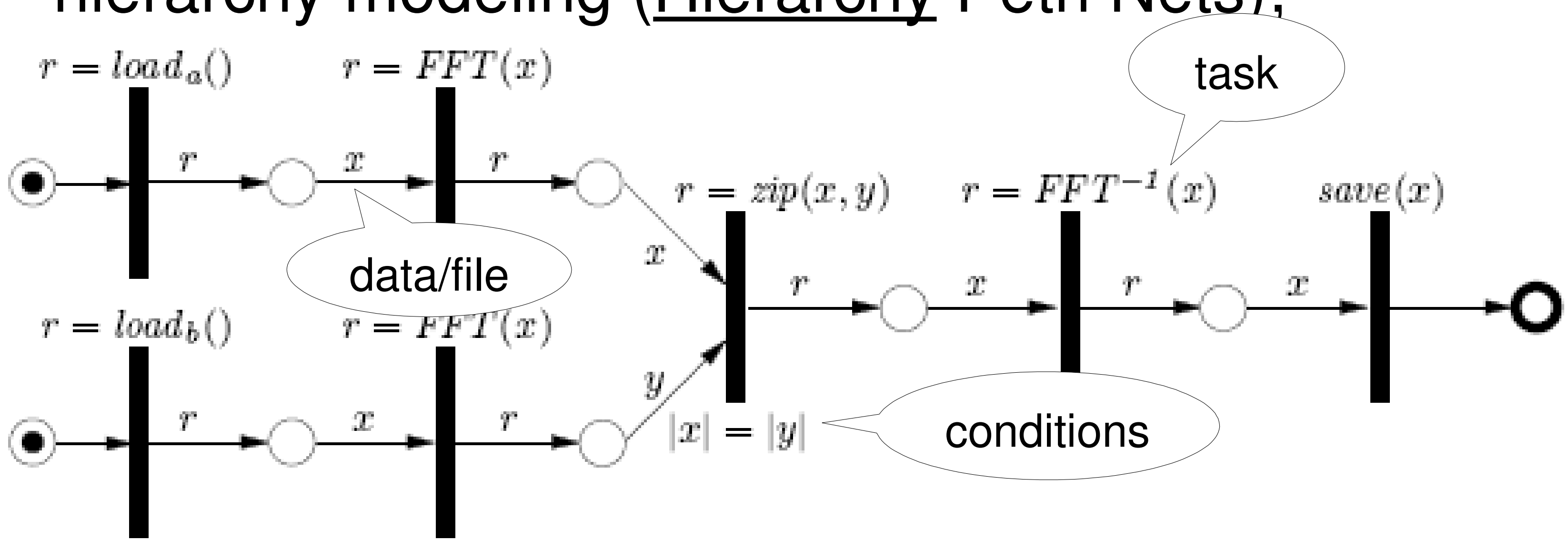
- **DEF:** *A transition t_j is live if it is potentially firable in all reachable markings* (as a consequence, a transition is live if it does not miss the possibility of fire)*;*
- Liveness is computationally *difficult to check*, so 5 different levels of liveness ($L_0 - L_4$) are defined;

Liveness (2/2)

- *A transition t_j in a Petri net is:*
  - *L_0 live, or dead, if and only if it can not be fired;*
  - *L_1 live if and only if it can possibly be fired;*
  - *...*
  - *L_4 live or simply live if and only if in any reachable state M, t_j is L_1 live* (equivalent to previous slide liveness definition).
- The concept of liveness is referred to the total absence of *deadlocks* in the net evolution;

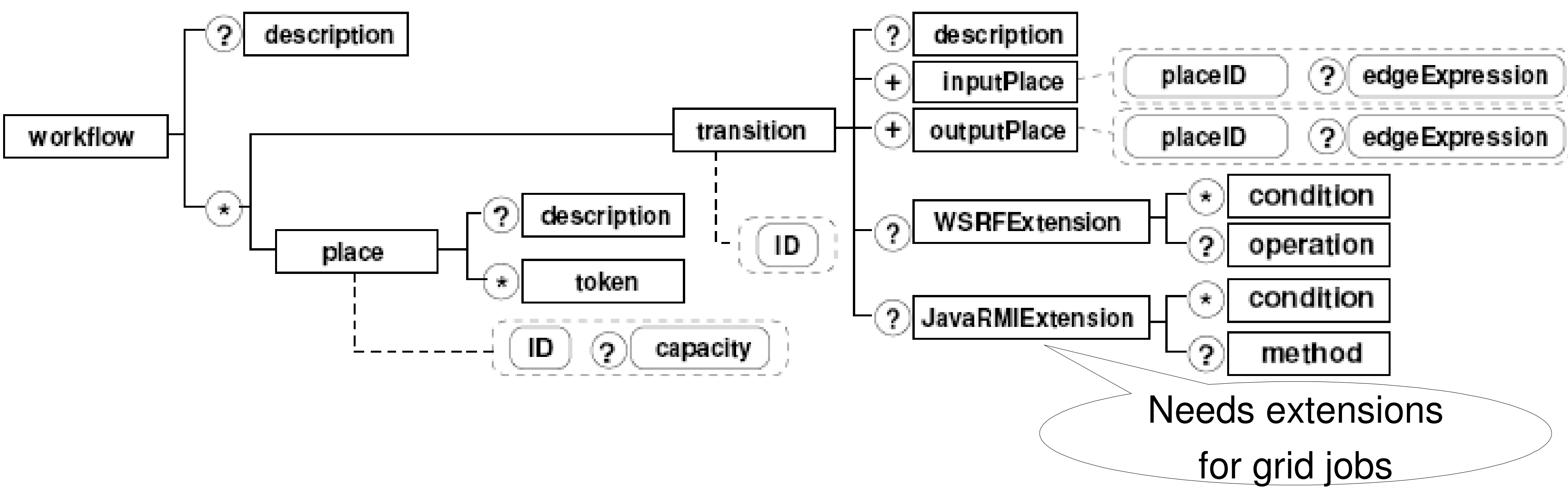
High Level Petri Net (HLPN)

- It is an extension to Petri net model (completely back compatible); it adds:
  - data modeling (Colored Petri Nets);
  - time modeling (Timed Petri Nets);
  - hierarchy modeling (Hierarchy Petri Nets);



Petri Net Representation

- XML-based language for representing Grid workflows based on HLPN;
- Describe either the *abstract* and the *concrete* (*extensions*) workflow;



END!