



# Combining Free choice and Time in Petri Nets

S Akshay, Loïc Hélouët, R Phawade

► **To cite this version:**

| S Akshay, Loïc Hélouët, R Phawade. Combining Free choice and Time in Petri Nets . 6th IFIP  
| Working group on trends in Concurrency, Sep 2017, Berlin, Germany. 2017. <hal-01650751>

**HAL Id: hal-01650751**

**<https://hal.inria.fr/hal-01650751>**

Submitted on 28 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combining Free choice and Time in Petri Nets

S.Akshay<sup>a</sup>, [L. Hélouët](#)<sup>b</sup>, R. Phawade<sup>a</sup>

<sup>a</sup>) IIT Bombay, India

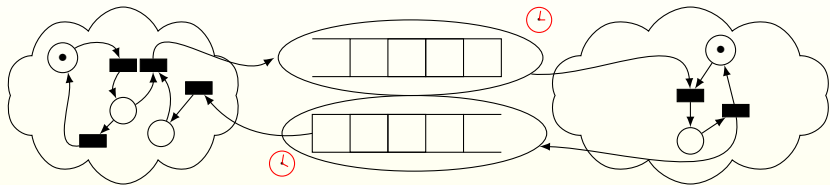
<sup>b</sup>) INRIA Rennes, France

TRENDS Sept. 9<sup>th</sup> 2017

[TIME'16]

# Motivation 1 : Modeling issues

Model Time constrained **unbounded** concurrent systems



## Desired features

- Latency : messages take **at least 10 ms** to reach their destination
- Timeout/urgency : a message not consumed **after 200ms** is lost
- Rates : a message is received **every  $\gamma$  t.u.,...**

# Motivation 2 : Verification

## Standard questions

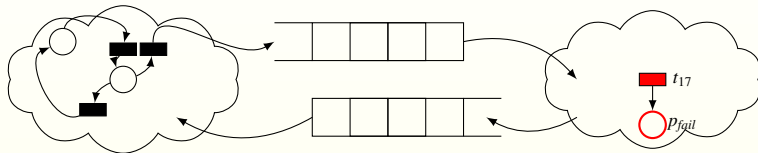
- **Reachability** : is marking  $M$  reachable from initial marking  $M_0$  ?
- **Coverability** : Given a marking  $M$ , is there a marking  $M'$  reachable from  $M_0$  such that  $M'(p) > M(p)$  for every place  $p$  ?
- **Boundedness** : is there a bound  $K$  such that for every reachable marking, every place  $p$ ,  $M(p) \leq K$  ?
- **Firability** : is there an execution in which transition  $t$  is fired ?

## Objectives

- Decidability for these questions
- Efficient algorithms

# Motivation 3 : Robustness

$\varphi_{17}$  : Transition  $t_{17}$  (a major failure) is not firable.



What if :

- time is measured with some imprecision
- clocks tend to have some drift/jitter/delay, ...

**Robustness : reasoning with idealized time representation**

Assume a class of properties  $\Phi$ , a model for time imprecision  $\llbracket \cdot \rrbracket_\delta$

Given a model  $\mathcal{M}$ , and a value  $\delta \in \mathbb{R}$ , check that :

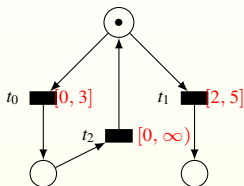
$$\forall \varphi \in \Phi, \mathcal{M} \models \varphi \iff \llbracket \mathcal{M} \rrbracket_\delta \models \varphi$$

Given a model  $\mathcal{M}$ , check if :

$$\exists \Delta, \forall \delta \leq \Delta, \forall \varphi \in \Phi, \mathcal{M} \models \varphi \iff \llbracket \mathcal{M} \rrbracket_\delta \models \varphi$$

# Time vs Timed Petri nets

*Time Petri nets* [Merlin74]

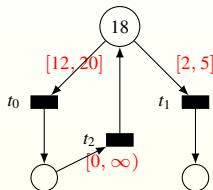


Pros :

- time
- Urgency
- unbounded places
- Expressive power

Cons : Undecidability

*Timed Petri nets* [Walter83]



Pros :

- time
- unbounded places
- Ages
- WSTS  $\equiv$  decidability of coverability, boundedness,...

Cons : no urgency

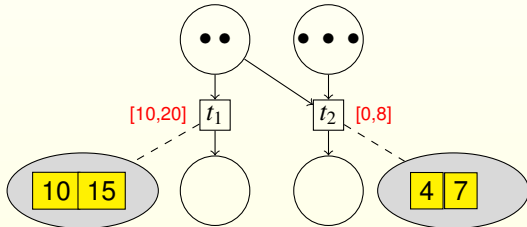
- Free-choice Multiserver Time Petri nets
- Processes and their relation to untimed nets
- Firability
- Robustness

# TPN Multiserver Semantics : Configuration

## Configuration (Threshold Semantics)

$$C = (M, mem)$$

- $M$  : marking, enables transitions **several** times
- $mem$  : remembers for each **enabling instance** of a transition for how long it has been enabled

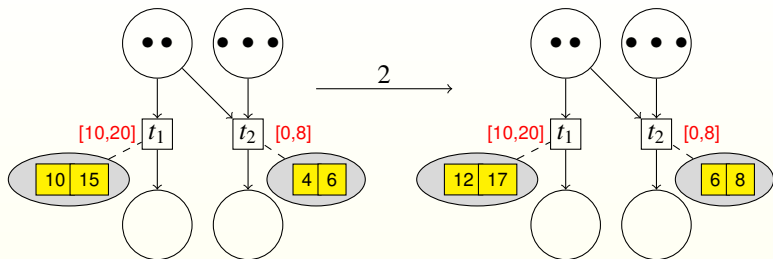


$$M(p_0) = 2; M(p_1) = 3; M(p_3) = 0; M(p_4) = 0$$

$$mem(t_1) = \{10; 15\} \quad mem(t_2) = \{4; 7\}$$



# TPN Multiserver Semantics : timed move



Timed Move :  $C \xrightarrow{\delta} C'$

Let a duration  $\delta$  elapse = update memorized durations

Urgency of a transition  $t$  in  $C = (M, mem)$

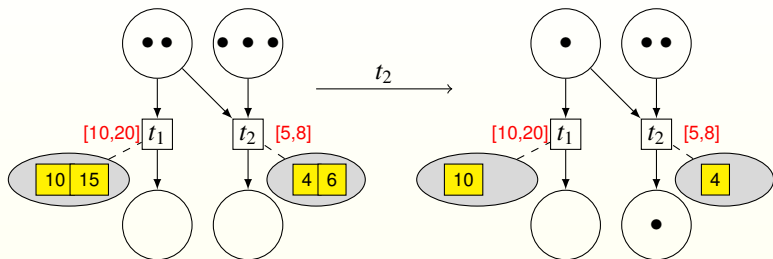
Let  $t$  be a transition such that :

- $I(t) = [l, u]$
- $t$  has been enabled for  $u$  t.u,                    i.e.,  $\max mem(t) = u$

then,  $t$  is **urgent**.

Time **cannot** progress, a **discrete** move must occur.

# TPN Multiserver Semantics : discrete move



Discrete firing :  $C \xrightarrow{t_i} C'$

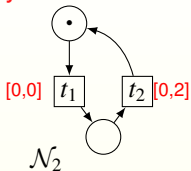
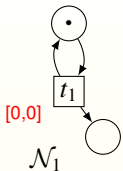
- $I(t_i) = [l, u], \max(\text{mem}(t_i)) > l$  ( $t_i$  need not be urgent)
- modification of memory for transitions in competition with  $t_i$
- Other enabling instances remain untouched

Behaviors of a net

- Labeled Transition System :  $(\mathcal{C}, \longrightarrow, C_0)$
- $\text{Lang}(\mathcal{N}) \subseteq T \times \mathbb{R}$  :  
set of timed words of  $\mathcal{N}$  :  $w = (t_1, d_1)(t_2, d_2) \dots$

# Restrictions (to obtain decidable classes)

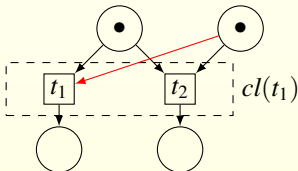
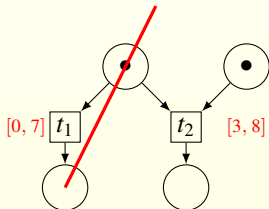
Forbid nets that can **force zero-delay behaviors**.



## Free choice PN and free choice TPN

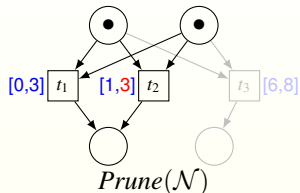
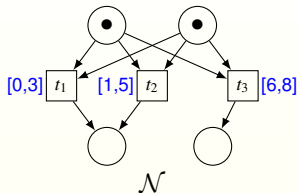
(FC-PN)  $\mathcal{U} = (P, T, F)$  is **free choice** if  $\forall t, t' \in T, \bullet t \cap \bullet t' \neq \emptyset \implies \bullet t = \bullet t'$ .  
(FC-TPN)  $\mathcal{N} = (\mathcal{U}, M_0, I)$  is a **free choice TPN** if  $Untime(\mathcal{N}) = \mathcal{U}$  is free choice.

The **cluster** of transition  $t$  is  $Cl(t) = \{t' \in T \mid \bullet t \cap \bullet t' \neq \emptyset\}$



# Pruning FC-TPN (a.k.a normalization [Chatain13])

Some transitions in FC-TPNs will obviously never fire !



To obtain a Pruned FC-TPN

- remove unfirable transitions :

$$t : I(t) = [a, b] \wedge \exists t', I(t') = [c, d] \text{ with } d < a$$

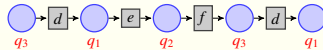
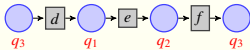
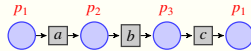
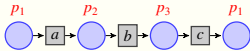
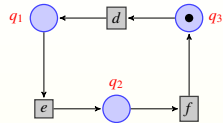
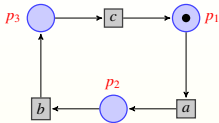
- Associate to remaining transition possible values for intervals

$$I(t) = [a, b] \Rightarrow I'(t) = [a, \max_{t' \in Cl(t)} (lft(t'))]$$

**Pruning Lemma :** Let  $\mathcal{N}$  be a **FC-TPN**, then

the transition systems associated with  $\mathcal{N}$  and  $Prune(\mathcal{N})$  are isomorphic.  
(not true outside FC-TPNs)

# Causal processes



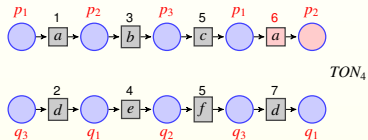
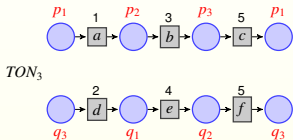
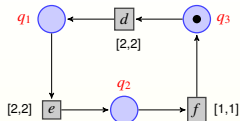
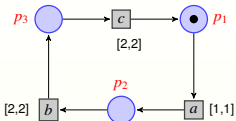
Untimed causal processes  $ON_1$  and  $ON_2$

## Causal processes

- A partial order representation of executions
- **Principle** : Unfold the net by glueing transitions/places occurrences one after another starting from the initially marked places

*Still an untimed setting !*

# Timed causal processes



## Advantages of Timed Causal processes

- Partial order timed representation of timed executions
- **URGENCY** is considered : if  $d$  occurs at date 7 in  $TON$ , all urgent transitions before  $d$  also occur in  $TON$ .

## Relation between Timed causal Processes and timed languages

Let  $\mathcal{P}$  be the set of timed causal processes of  $\mathcal{N}$ . Then,

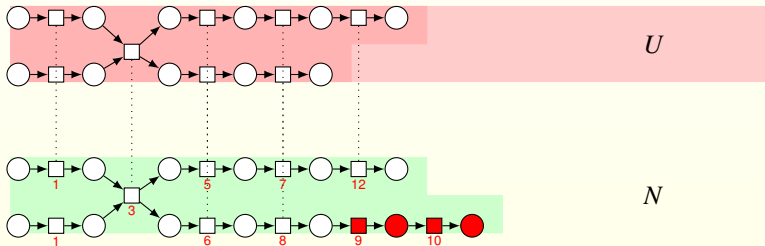
$$Lang(\mathcal{N}) = \bigcup_{N \in \mathcal{P}} Lang(N)$$

# Properties of Free Choice TPN

## Theorem 1 : Inclusion of untimed prefixes

Let  $\mathcal{N} = (\mathcal{U}, M_0, I)$  be a **pruned FC-TPN** (w.o. forced 0-delay sequences).  
 $U$  be an (untimed) causal process of  $\mathcal{U} = \text{Untime}(\mathcal{N})$ .

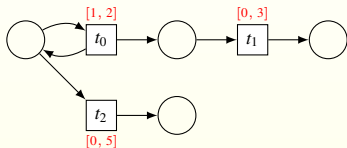
Then there exists a timed causal process  $N$  of  $\mathcal{N}$  such that  $U \leq \text{Untime}(N)$ .



## Theorem 2 : Fireability

Let  $\mathcal{N} = (\mathcal{U}, M_0, I)$  be an FC-TPN (w.o. forced 0-delay).  
Checking fireability of a transition  $t \in T$  in  $\mathcal{N}$  is **decidable**.

Proof idea : Firability  $\sim$  coverability in untimed nets [Rack78]



## Theorem 3 : Termination

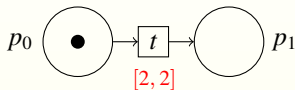
Let  $\mathcal{N} = (\mathcal{U}, M_0, I)$  be an FC-TPN (w.o. forced 0-delay). It is decidable if  $\mathcal{N}$  terminates.

### **Proof Idea :**

an infinite run of  $\mathcal{U} = \text{Untime}(\text{Prune}(\mathcal{N}))$  has a timed counterpart in  $\mathcal{N}$  and conversely. (and termination is decidable in untimed Petri nets)

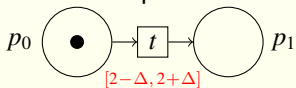


A major drawback of timed models :

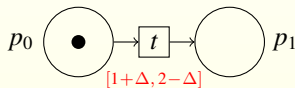


```
public class MyProgram {  
    throws InterruptedException {  
        Myprogram prog=new MyProgram();  
        //Pause for 2 seconds  
        Thread.sleep(2000);  
        prog.t();  
    }  
}
```

What the implementation might really do :



*Guard enlargement*



*Guard Shrinking*

Form now :  $\mathcal{N}_\delta = \mathcal{N}$  with enlarged guards

## Definition (robustness problems for TPNs)

Given a TPN  $\mathcal{N}$ , does there exist  $\Delta \in \mathbb{Q}_{>0}$  such that  $\forall \delta \leq \Delta$

- $\text{Fireable}(\mathcal{N}) = \text{Fireable}(\mathcal{N}_\delta)$ ?
- $\mathcal{N}$  terminates iff  $\mathcal{N}_\delta$  terminates
- $\mathcal{N}$  is bounded iff  $\mathcal{N}_\delta$  is bounded...

Note : Firability, termination, boundedness are not a priori robust/non robust properties of the whole class of FC-TPNs.

## Theorem 4

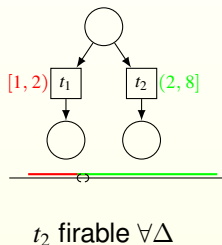
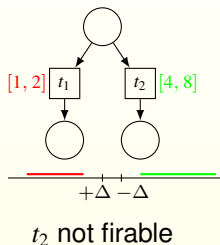
Let  $\mathcal{N}$  be a FC-TPN without forced 0-delay time firing sequences. Then robustness of fireability is decidable. If  $\mathcal{N}$  has robust fireability, bound  $\Delta$  can be effectively computed.

## Theorem 5

Let  $\mathcal{N}$  be a FC-TPN without forced 0-delay time firing sequences. Then it is decidable whether termination is a robust property of  $\mathcal{N}$ .

# Proof idea

If  $\text{Prune}(\mathcal{N})$  and  $\text{Prune}(\mathcal{N}_\Delta)$  have the same clusters, then  $\Delta$ -enlargement of  $\mathcal{N}$  does not modify firable transitions.



Check for each accessible cluster  $C$  whether intervals can be enlarged by some  $\Delta_C$  without changing firable transitions

Contributions so far :

- A FC-Multiserver PN variant with its process semantics
- Decidable firability, termination
- Decidability of robustness of firability, termination, wrt enlargement (and shrinking)

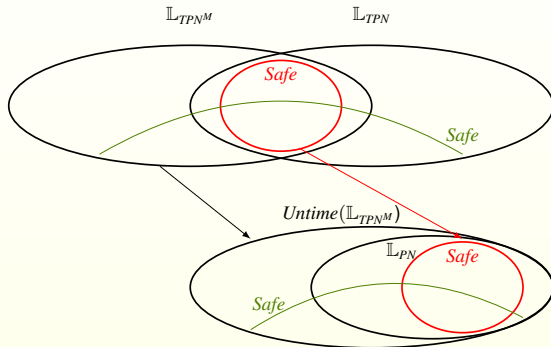
Open questions :

- decidability of Coverability, reachability and boundedness for FC-TPNs ?
- Robustness of more properties ?
- same issues without multi-enabledness ?
- Expressiveness of FC-TPNs ?

$\mathbb{L}_{TPNM}$  : Timed languages expressible with Time Petri nets (Multiserver)

$\mathbb{L}_{TPN}$  : Timed languages expressible with Time Petri nets

$\mathbb{L}_{PN}$  : Untimed languages expressible with Petri nets



(Thm. 1) In free choice nets, timed / untimed processes tightly related  
What about **Languages** ?

$$\mathbb{L}_{PN} \subseteq Untime(\mathbb{L}_{TPNM}) \quad \mathbb{L}_{PN \cap Safe} = Untime(\mathbb{L}_{TPNM \cap Safe}) \quad \mathbb{L}_{PN} \neq Untime(\mathbb{L}_{TPNM} ??)$$

# References

Time & PN related Publis (SUMO group, INRIA Rennes) :

[AHP16] S. Akshay, L. Hélouët, R. Phawade, *Combining free choice and time in petri nets*, IEEE Proc. of TIME 2016, pp. 120–129, 2016.

[AHP17] S. Akshay, L. Hélouët, R. Phawade, *Combining free choice and time in petri nets (Extended Version)*,

<http://people.rennes.inria.fr/Loic.Helouet/Papers/Lamp.pdf>

[AGH16] S. Akshay, B. Genest, L. Hélouët, *Decidable classes of unbounded Petri nets with time and urgency*, in : PETRI NETS'16, Vol. 9698 of LNCS, Springer, 2016, pp. 301–322.

[AHJR16] S. Akshay, L. Hélouët, C. Jard, P.-A. Reynier, *Robustness of time Petri nets under guard enlargement*, Fundam. Inform. 143 (3-4) (2016) 207–234.

Petri nets & semantics :

[Merlin74] P. Merlin, *A study of the recoverability of computing systems*, Ph.D. thesis, University of California, Irvine, CA, USA (1974).

[EsparzaD95] J. Esparza, J. Desel, *Free Choice Petri nets*, Cambridge University Press, 1995.

[BoyerD01] M. Boyer, M. Diaz, *Multiple enabledness of transitions in Petri nets with time*, in : Proc. of PNPM'01, IEEE, 2001, pp. 219–228.

[AuraL00] T. Aura, J. Lilius, *A causal semantics for time Petri nets*, TCS 243 (1-2) (2000) 409–447.

[Chatain13] T. Chatain, C. Jard, *Back in time Petri nets*, in : Proc. of FORMATS'13, Vol. 8053 of LNCS, Springer, 2013, pp. 91–105.

# Bibliography (continued)

## Verification of Concurrent systems :

[Jones77] N. Jones, L. Landweber, Y. Lien, *Complexity of some problems in Petri nets*, TCS 4 (3) (1977) 277–299.

[AbdullaN01] P. Abdulla, A. Nylén, *Timed Petri nets and BQOs*, in : Proc. of ICATPN 2001, Vol. 2075 of LNCS, Springer, 2001, pp. 53–70.

Rack78 C. Rackoff, *The covering and boundedness problem for vector addition systems*, TCS 6 (1978) 223–231.

[FinkelL15] A. Finkel, J. Leroux, *Recent and simple algorithms for Petri nets*, Software and System Modeling 14 (2) (2015) 719–725.

[Hack76] Hack, M. : *Decidability Questions for Petri Nets*, Ph.D. Thesis, M.I.T., MIT, CA, USA, 1976.

[KarpM69] Karp, R., Miller, R. : *Parallel program schemata*, In *JCSS*, **3**, 1969, 147–195.

[CHSS13] L. Clemente, F. Herbreteau, A. Stainer, G. Sutre, *Reachability of communicating timed processes*, in : FoSSaCS, Vol. 7794 of LNCS, 2013, pp. 81–96.

# Bibliography (continued)

## Robustness

[Puri00] A. Puri, *Dynamical properties of timed automata*, In DEDS 10 (1-2) (2000) 87–113.

[BouyerMS11] Bouyer, P., Markey, N., Sankur, O. *Robust Model-Checking of Timed Automata via Pumping in Channel Machines*, Proc. of FORMATS, 6919, Springer, 2011.

[DDMR08] De Wulf, M., Doyen, L., Markey, N., Raskin, J.-F. : *Robust Safety of Timed Automata*, Formal Methods in System Design, **33**(1-3), 2008, 45–84.

[DDR05] De Wulf, M., Doyen, L., Raskin, J.-F., *Systematic Implementation of Real-Time Models*, Proc. of Formal Methods, 3582, Springer, 2005.

[Sankur11] Sankur, O., *Untimed Language Preservation in Timed Systems*, Proc. of MFCS, 6907, Springer, 2011.

[SwaminathanFK08] Swaminathan, M., Fränzle, M., Katoen, J.-P., *The Surprising Robustness (Closed) Timed Automata against Clock-Drift*, Proc. of TCS, Springer, 2008