# Refinement of time Petri nets semantics in conflict situations

Adrien Bullich, Hanifa Boucheneb and Olivier H. Roux,

***Abstract*—This paper deals with time Petri nets, where a firing interval is associated with each transition. Three semantics (intermediate, atomic and persistent atomic) are proposed, in the literature for this model, in the context of single-server/multi-server and strong/weak semantics. This paper shows that, in presence of conflicts, these semantics may exhibit some unexpected behaviours and properties. This paper proposes a new semantics more appropriate to deal with conflicts.**

***Index Terms*—Time Petri nets, semantics, conflicts, zone based graph**

## I. Introduction

Increasing complexity of systems used nowadays requires rigorous formalisms and tools to automatically verify and control their behaviours. From this perspective, several formalisms such as Petri nets, automata and logics have been developed. Their purpose is to represent, using mathematical concepts, systems in order to be able to verify and control the conformity of their behaviour w.r.t. their expected services.

Automata and Petri nets are designed to model discrete systems. In the context of real time systems, where the behaviour is dependent of time, the used formalisms must integrate explicitly the time factor. Timed automata and time Petri nets appear in order to model hybrids systems, handling discrete systems with continuous variables, i.e. the time. Hybrids systems model both the process and the control system.

Many ways exist to consider time in Petri nets. The time constraints may be expressed in terms of stochastic delays of transitions (stochastic Petri nets), fixed values associated with places or transitions ({P,T}-Timed Petri nets) [13], or intervals labeling places, transitions or arcs ({P,T,A}-Time Petri Nets) [6], [9]–[11], [14].

For {P,T,A}-Time Petri Nets, there are two firing semantics: Weak Time Semantics (WTS) and Strong Time Semantics (STS). For both semantics, each enabled transition has an explicit or implicit firing interval derived from time constraints associated with places, transitions or arcs of the net. A transition cannot be fired outside its firing interval, but in WTS, its firing is not forced when the upper bound of its firing interval is reached. Whereas in STS, it must be fired within its firing interval unless it is disabled. The STS is the most widely used semantics. There are also multiple-server and single-server semantics. The multiple-server semantics allows to handle, at the same time, several time intervals per place (P-TPN), per arc (A-TPN) or per transition (T-TPN), which it is not allowed in the single-server semantics.

In this paper, we consider T-time Petri nets (Merlin's model) [10], called here time Petri nets (TPN in short), in the context of single-server and strong semantics. It seems to be strongly appropriate with communications protocol. In this model, a time interval is associated with each transition. From the semantic point of view, a clock is associated with each transition to measure its enabling time. A transition is firable if its clock has reached its interval and must be fired before overpassing its interval, unless it is disabled by another firing. In [2], three semantics intermediate, atomic and persistent atomic are discussed for time Petri nets. They differ in the way that clocks are handled (memory policies), when a transition is fired. The intermediate semantics resets clocks of all transitions disabled when input tokens of the fired transition are consumed (intermediate marking). The atomic and persistent semantics suppose that the firing of a transition is atomic and do not consider the intermediate marking.

In general, the intermediate semantics is weakly expressive, w.r.t. the weak timed bisimulation, in comparison with the atomic and the atomic persistent ones [2]. But, for time Petri nets with upper-closed intervals[1], the three semantics are equivalent w.r.t. the weak timed bisimulation [2]. From the practical point of view, the atomic and persistent atomic semantics are more appropriate for the specification of observers of systems [2]. The intermediate one seems to be closer to the intuitive interpretation and, for this reason, is widely used. By intuitive interpretation (or semantics) of time Petri nets, we mean that a transition may be fired if it is maintained continuously enabled (using the same tokens) until reaching its firing interval (i.e., its enabling time is inside its firing interval).

In this paper, we first show that, in presence of conflicts, these semantics may exhibit some unexpected behaviours and properties w.r.t. the intuitive semantics. Then, we propose a new semantics more appropriate to deal with conflicts.

This paper is organized as follows. Section II is devoted to the definition of Merlin's model [10] and a short review of the different semantics proposed in the literature for this model. In Section III, we show, by means of examples, that in some conflicting situations, the scenarios obtained w.r.t. the intermediate semantics doesn't respect the constraint of waiting time. In Section IV, we propose a new semantics

A. Bullich and O. H. Roux are with laboratory IRCCyN, 1, rue de la Noë, 44 300 Nantes, France.
E-mail: adrien.bullich@irccyn.ec-nantes.fr, olivier-h.roux@irccyn.ec-nantes.fr

H. Boucheneb is with the Department of Computer Engineering and Software Engineering, École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal, Québec, Canada, H3C 3A7, Québec, Canada.
E-mail: hanifa.boucheneb@polymtl.ca

[1] An upper-closed interval is an interval of the form $[a, b], ]a, b], [a, \infty[$ or $]a, \infty[$.

whose idea is to measure the waiting time of each token. Finally, the conclusion is presented in section V.

## II. TIME PETRI NETS AND THEIR SEMANTICS

### A. *Timed transition system*

Usually the semantics of a timed model is defined by means of a timed transition system, where the set of states of the model, its actions as well as its transition relation between states are defined. The actions of a timed model are of two types: discrete actions for events and positive real numbers for time elapsing.

Formally, a timed transition systems is defined by a 4-uplet $< Q, q_0, \Sigma, \rightarrow >$ where $Q$ is a set of states, $q_0 \in Q$ is the initial state, $\Sigma$ is the set of discrete actions (disjoint from the time domain $\mathbb{R}^+$ of the continuous actions), and $\rightarrow \in Q \times (\Sigma \cup \mathbb{R}^+) \times Q$ is the transition relation. A tuple $(q, a, q') \in \rightarrow$, also denoted $q \xrightarrow{a} q'$, represents the transition from state $q$ to state $q'$ by the discrete or continuous action (time progression) $a$.

### B. *Definition of TPN*

Petri nets, introduced by Petri in 1962, with their useful abbreviations and extensions are a powerful formalism, which allows precise modeling and analysis of complex systems, using a wide range of methods and tools.

This paper deals with time Petri nets, a simple yet powerful model useful to model and verify real time systems, like communications protocol. This model associates a firing interval with each transition. It allows to model different kinds of time constraints (delays, durations, deadlines, etc.), even if the exact delays or durations of events are not known. Formally, a TPN is defined by a 7-uplet: $< P, T, Pre, Post, \alpha, \beta, M_0 >$ where:

- $P$ is the set of places in the net;
- $T$ is the set of transitions (s.t. $P \cap T = \varnothing$);
- $Pre \in [P \times T \longrightarrow \mathbb{N}]$ is the backward incidence function, indicating, for each transition, the tokens needed for its firing;
- $Post \in [P \times T \longrightarrow \mathbb{N}]$ is the forward incidence function, indicating, for each transition, the tokens produced by its firing (we denote $C = Post - Pre$ the incidence function);
- $\alpha \in [T \rightarrow \mathbb{Q}^+]$ is a function which associates with each transition the lower bound of its firing interval;
- $\beta \in [T \longrightarrow \mathbb{Q}^+ \cup \{\infty\}]$ is a function which associates with each transition the upper bound of its firing interval;
- $M_0 \in [P \longrightarrow \mathbb{N}]$ is the initial distribution of tokens in places, called the initial marking.

For convenience, we denote $t^\bullet = \{p \in P | Post(p, t) > 0\}$ the set of output places of $t$ and ${}^\bullet t = \{p \in P | Pre(p, t) > 0\}$ the set of input places of $t$. We suppose here that ${}^\bullet t \neq \emptyset$, for every transition of the net.

A time Petri net evolves according to two aspects: the marking and clocks. Thus, we can represent the global state of a time Petri net by a pair $(M, \nu)$, where $M \in [P \longrightarrow \mathbb{N}]$ is a marking of the Petri net, and $\nu \in [T \longrightarrow \mathbb{R}^+]$ is a clock valuation over $T$, which associates with each transition, the value of its clock. Its initial state is $(M_0, \mathbf{0}_T)$, where $M_0$ is the initial marking and $\mathbf{0}_T$ is the null valuation over $T$.

A transition $t$ is enabled in $M$, if there are enough tokens in $M$ for its firing (that means $M \geq Pre(., t)$). The firing of $t$ takes no time but leads to the marking $M'$ obtained by consuming tokens of $Pre(., t)$ and producing tokens of $Post(., t)$: $M' = M - Pre(., t) + Post(., t)$.

In time Petri nets, a transition is firable at state $(M, \nu)$ iff it is enabled and its clock has reached its associated interval. We denote $firable(M, \nu)$ the set of transitions firable at state $(M, \nu)$: $firable(M, \nu) = \{t \in T | M \geq Pre(., t) \wedge \nu(t) \in [\alpha(t), \beta(t)]\}$.

When a transition $t$ is fired, a new marking is reached, where we can find some newly enabled transitions. We denote $\uparrow enabled(M, t)$ the set of transitions newly enabled in the marking reached from $M$ by firing the transition $t$. It indicates the clocks that are reseted when $t$ is fired from $M$.

In time Petri nets, all clocks of transitions evolve uniformly with time. We denote $\nu + d$ the function $\nu'$ such that $\forall t \in T, \nu'(t) = \nu(t) + d$. It specifies the evolution of time by $d$ units.

The behaviour of a time Petri net is defined by means of the following timed transition system $< Q, q_0, \Sigma, \rightarrow >$, where $Q = (P \longrightarrow \mathbb{N}) \times (T \longrightarrow \mathbb{R}^+)$ is the set of states of the time Petri net, $q_0$ is its initial state, $\Sigma = T$, and $\rightarrow$ is composed of continuous and discrete transitions defined as follows:
Let $(M, \nu)$ be a state, $d \in \mathbb{R}^+$, $t \in T$, $M'$ a marking and $\nu'$ a clock valuation over $T$.

- Continuous transition:
$$(M, \nu) \xrightarrow{d} (M, \nu') \text{ if}$$
$$\begin{cases} \nu' = \nu + d \\ \forall t \in T : t \notin firable(M, \nu + d) \Rightarrow \\ \qquad (\forall d' \in [0, d] : t \notin firable(M, \nu + d')) \end{cases}$$
- Discrete transition:
$$(M, \nu) \xrightarrow{t} (M', \nu') \text{ if}$$
$$\begin{cases} t \in firable(M, \nu) \\ M' = M - Pre(., t) + Post(., t) \\ \forall t' \in T : \nu'(t') = \begin{cases} 0 \text{ if } t' \in \uparrow enabled(M, t), \\ \nu(t') \text{ otherwise.} \end{cases} \end{cases}$$

### C. *Semantics of TPN in the memory policy*

Different semantics can be derived from the transition system given above, depending on the definition of the notion of newly enabled (memory policy) [1], [2]. The definition of this notion has an impact on the behaviour and the properties of the net. In [2], the authors have distinguished three memory policies: intermediate, atomic and persistent atomic semantics.

*a) Intermediate semantics:* In the intermediate semantics, the firing of a transition consists of two steps: consuming tokens and producing tokens. A distinction is then made between tokens used by a transition and those produced. All transitions not enabled in the marking resulting from the first step (intermediate marking) but enabled, in the marking resulting from the second step, are newly enabled. In other words, let $M$ be a marking, $t$ and $t'$ two transitions. The transition $t'$ is (newly) enabled by firing $t$ from $M$, if in the intermediate marking of this firing, $t'$ is disabled ($Pre(., t') \not\leq M - Pre(., t)$), but is enabled after the firing of $t$ ($Pre(., t') \leq M + C(., t)$). Moreover, in the context of single-server semantics, the intermediate semantics resets the clock of the fired transition ($t = t'$). Therefore, the set of the newly enabled transitions by firing $t$ from $M$ is defined by:

$$\uparrow enabled(M,t) = \{t' \in T | Pre(.,t') \leq M + C(.,t) \wedge$$

$$(Pre(.,t') \nleq M - Pre(.,t) \ \vee \ t = t')\}$$

*b) Atomic semantics:* In the atomic semantics [2], the firing of a transition is supposed atomic. Unlike the intermediate semantics, the atomic one does not consider the intermediate markings. In this case, all transitions not enabled before firing a transition but enabled after its firing are newly enabled. More precisely, let $M$ be a marking, $t$ and $t'$ two transitions. The transition $t'$ is (newly) enabled by firing $t$ from $M$, if it belongs to the following set:

$$\uparrow enabled(M,t) = \{t' \in T | Pre(.,t') \leq M + C(.,t) \wedge$$

$$(Pre(.,t') \nleq M \ \vee \ t = t')\}$$

*c) Persistent atomic semantics:* In the persistent atomic semantics [2], a transition $t'$ is newly enabled after the firing of $t$ in the marking $M$ if it belongs to the set:

$$\uparrow enabled(M,t) = \{t' \in T | Pre(.,t') \leq M + C(.,t) \wedge$$

$$Pre(.,t') \nleq M\}$$

The difference between the atomic semantics and the persistent atomic one lies in the particular case of the fired transition. If the fired transition enables again itself, its clock is reseted in the atomic semantics but not reseted in the persistent atomic semantics. It is then considered as a newly enabled transition in the atomic semantics but not newly enabled in the persistent atomic semantics.

### D. Age or threshold semantics

Boyer, in [5], considers two kinds of semantics, according to the meaning of clocks. In the first one, the *age semantics*, the firing condition is about the waiting time of tokens. When a transition is labelled $[a; b]$, it means that tokens must wait between $a$ and $b$ time unit to fire. It is the case of the machining, for example. In the second one, the *threshold semantics*, tokens are not distinguished, only matters the load. When a transition is labelled $[a; b]$, it means that the number of tokens must be greater than the weight during $a$ to $b$ time unit to fire the transition. This conceptualisation is adapted to load mechanisms.

Different models with age semantics exist in the literature, like P-time Petri nets and A-time Petri nets. In these two models, the fire of a transition considers the age of tokens and not a number. However, it seems less used in T-time Petri nets. We'll see for example that intermediate, atomic and persistent atomic semantics are threshold semantics.

## III. Problem of handling conflicts

The way that conflicts are handled differs from one semantics to the other, leading to different behaviours and properties. So, it is essential for a semantics to be clearly and coherently defined, in order to avoid incoherences, in the manner that similar situations are managed. To be intuitive is a true advantage for a semantics.

The intermediate semantics distinguishes tokens, thanks to the intermediate step, between produced tokens and the others. However, it is important to note that the intermediate semantics is not an age semantics. That's the matter of our first study: what difference we can find in case of conflicts.

Let us explain, by means of examples, such conflicting situations and their impact on the behaviour and properties of the model.

### A. Change of behaviour

Consider the net at Fig. 1. According with the waiting time of tokens, the system is expected to behave as follows: the token in place $P1$ goes to the place $P2$ at date 1. The initial token in $P2$ is either consumed by $T2$ or $T3$ at date 1. The token created by $T1$ in $P2$ should be consumed at date 2.

Suppose now that the transition $T1$ is fired before the others, from the initial marking $P1 + P2$. This firing leads to the marking $2P2$, where both transitions $T2$ and $T3$ are enabled and not in conflict. These two transitions were enabled but in conflict in the initial marking (before firing $T1$). Let us examine how the three memory policies handle this situation.

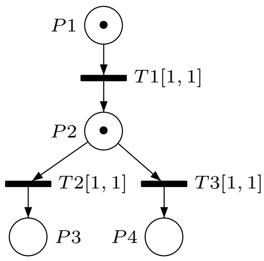


Fig. 1. A TPN with two conflicting transitions

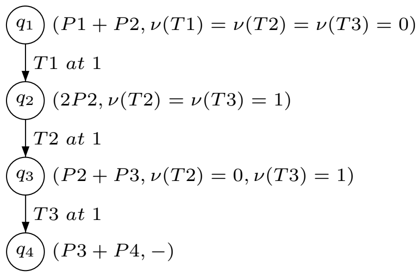


Fig. 2. An unexpected run of the TPN at Fig. 1

In the intermediate semantics, when the transition $T1$ is fired from the state $(P1+P2, \nu(T1) = \nu(T2) = \nu(T3) = 1)$, clocks of both transitions $T2$ and $T3$ are not reseted, since they are enabled before, during and after firing $T1$. With this semantics, the firing of $T1$ leads to the state $(2P2, \nu(T2) = \nu(T3) = 1)$. From this state, the model behaves as if both tokens in $P2$ were created at the same time (at date 0). For instance, if the first token in $P2$ is used by $T2$, then $T3$ is considered as not newly enabled and fired immediately after $T2$ (see Fig. 2). Unlike what is expected, the two tokens reach their destination at the same time. Transition $T3$ is fired even if its token is just created and did not wait the one time unit needed. The constraint of waiting time is then not respected for $T3$. From this point of view, this semantics seems to be incoherent with the context of age semantics.

Both the atomic and persistent atomic semantics accept the same unexpected run.

## B. Change of properties

Let us now show that the change in managing conflict, pointed out in the intermediate semantics, may have an impact on the properties of the model. Consider the TPN at Fig. 3. The TPN is bounded w.r.t. the intermediate semantics but is unbounded w.r.t. the intuitive semantics.

In this net, the role of the transition $T3$ should be to empty the place $P4$ and then prevent the system to reach a state where $P5$ is marked. From such a state, the transition $T5$ will be repeatedly fired every one time unit, leading to an infinite number of markings (unbounded net).

In the intermediate semantics, from the initial state $(P1 + P2 + P4, \nu(T1) = \nu(T2) = \nu(T3) = 0)$, the model can fire the transition $T1$ at date 2 to reach the state $(2P2 + P4, \nu(T2) = \nu(T3) = 2)$. From this state, $T2$ and $T3$ are fired successively at dates 3 and 4, leading to the dead marking $P3$ (see Fig.4). Therefore, the enabling time constraint is not respected for $T3$. Indeed, in this scenario, the transition $T3$ uses the token created by $T1$ and then should be fired 4 time units after $T1$ (at date 6), unless it is disabled by firing a conflicting transition. After firing successively transitions $T1$ and $T2$ at dates 2 and 3, both transitions $T3$ and $T4$ are enabled but in conflict. They are both firable at date 6. The firing of $T3$ will disable $T4$ and mark the place $P5$ and then enable $T5$. The model is then unbounded w.r.t. the intuitive semantics.

Note that the run given at Fig. 4 is also valid in the context of the atomic semantics. For the persistent atomic semantics, the accepted run coincides with the one given at Fig. 4 until firing $T2$. In the persistent atomic semantics, as the firing of $T2$, at date 3, enables again $T2$. Its clock is then not reseted and the state reached by $T2$ is $(P2 + P3 + P4, \nu(T2) = \nu(T3) = 3, \nu(T4) = 0)$. From this state, the transition $T2$ is fired again at date 3, which disables $T3$. The reached state is then $(2P3 + P4, -)$.

We have shown that for the same net, the way that conflicting transitions are managed may have an impact on the behaviour and the properties of the model. It is then very important to understand how the different semantics handle the subtle cases of conflicts and to be sure that the behaviour of the model, w.r.t. a given semantics, corresponds exactly to the expected behaviour.

Note that a TPN can be unbounded w.r.t. the intermediate, atomic or persistent atomic semantics but bounded w.r.t. an intuitive semantics about age of tokens. As an example the TPN at Fig. 5 is unbounded w.r.t. the intermediate semantics but bounded w.r.t. the intuitive semantics. Therefore, there is no relationship between properties of the model w.r.t. the intuitive semantics and the others.

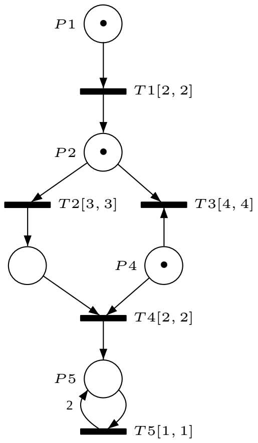


Fig. 3. A bounded TPN w.r.t. the intermediate semantics

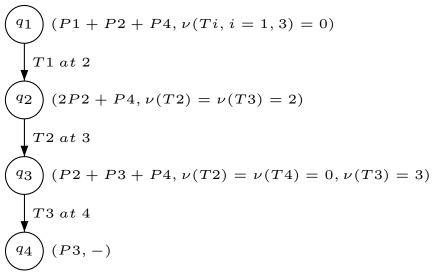


Fig. 4. A run of the TPN at Fig. 3 w.r.t. the intermediate semantics

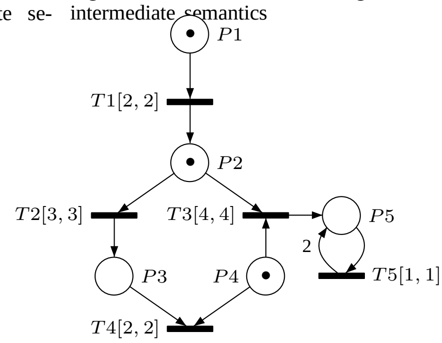


Fig. 5. An unbounded TPN w.r.t. the intermediate semantics

## IV. A TOKENS SEMANTICS

The enabling time of the transition is different with the age of the youngest tokens used by the transition. To deal with this difference, we need to identify tokens used by each transition and to memorize the age of tokens, as it is evokated in [5]. Doing so, we can make sure that the enabling time of each transition refers to the tokens to be consumed by the transition.

Moreover, in the context of single-server semantics, in the TPN, only one clock is associated with each transition. This clock is used to measure the time elapsed since it was last enabled. The different enabling instances of the same transition are handled sequentially.

To manage well the allocation of tokens to transitions w.r.t. single-server semantics, we associate a clock with each token and a queue of clock values with each place. The tokens of each place $p$ are then handled according to the FIFO (First In First Out) discipline (the first token created in $p$ is the first token consumed from $p$).

When a token is created in a place $p$, its clock is set to $0$. This value is inserted in the queue of $p$. All clocks of tokens evolve synchronously with time until they are consumed. An enabled transition is firable if the age of the youngest tokens participating in its enabling has reached the firing interval of the transition. It must be fired without any additional delay, if the age of its youngest tokens has reached the upper bound of its firing interval, unless it is disabled.

### A. Formalisation

Formally, we define the TPN state by a pair $(M, \mu)$, where $M$ is a marking and $\mu$ is a function over $P$, which associates with each place $p$ a queue of clock values. Each queue is managed FIFO and then ordered from the older to the younger (decreasing order of ages). For example, if a place $p$ has 4 tokens with ages $3, 3, 2$ and $1$, its queue is $\mu(p) = [3, 3, 2, 1]$.

The initial state is $(M_0, \mu_0)$, where $\forall p \in P, \mu_0(p)[i] = 0$, for $1 \leq i \leq M_0(p)$, if $M_0(p) > 0$, and $\mu_0(p) = []$, otherwise.

As mentioned before, tokens within each place are handled FIFO, an enabled transition will always use the oldest tokens from each place. Thus, if an enabled transition $t$ uses $Pre(p,t)$ tokens from the place $p$, the age of the youngest token from $p$ used by $t$ is the element $\mu(p)[Pre(p,t)]$ of the queue $\mu(p)$. For example, if $\mu(p) = [3,3,2,1]$ and $t$ needs 3 tokens from $p$, then it uses 2 tokens with age 3, and 1 token with age 2. The age of its youngest token is exactly its enabling time (i.e., 2).

Let $(M,\mu)$ be a state and $t$ a transition enabled in $M$. We define the enabling time, the firing condition and the set of firable transitions as follows:

- The enabling time of $t$ is the age of the youngest token used by $t$:
$$\min_{p \in {}^\bullet t} \{ \mu(p)[Pre(p,t)] \}.$$
- The firing condition of $t$ is then:
$$\min_{p \in {}^\bullet t} \{ \mu(p)[Pre(p,t)] \} \in [\alpha(t), \beta(t)].$$
- The set of firable transitions from a state $(M,\mu)$ is:
$$firable(M,\mu) = \{t \in T | M \geq Pre(.,t) \wedge \min_{p \in {}^\bullet t} \{ \mu(p)[Pre(p,t)] \} \in [\alpha(t), \beta(t)]\}$$

Formally, our semantics is defined by the transition system $< Q, (M_0, \mu_0), T, \rightarrow >$ where $Q = ([P \longrightarrow \mathbb{N}]) \times ([P \longrightarrow Queues])$ is the set of states and the transition relation $\rightarrow$ is defined as follows: Let $(M,\mu)$ and $(M',\mu')$ be two states, $t$ a transition of $T$ and $d$ a nonnegative real number ($d \in \mathbb{R}^+$).

- Continuous transition:
$(M,\mu) \xrightarrow{d} (M',\mu')$ if[2]
$$\begin{cases} M' = M \\ \mu' = \mu + d \\ \forall t \notin firable(M, \mu + d) \Rightarrow \\ \quad \forall d' \in [0;d[ t \notin firable(M, \mu + d') \end{cases}$$
- Discrete transition:
$(M,\mu) \xrightarrow{t} (M',\mu')$ if
$$\begin{cases} t \in firable(M,\mu) \\ M' = M - Pre(.,t) + Post(.,t) \\ \mu' \text{ is built this way :} \\ \text{Initialise } \mu' \text{ with } \mu; \\ \forall p \in^\bullet t, \text{ pop from } \mu'(p) \text{ the } Pre(p,t) \text{ first elements;} \\ \forall p \in t^\bullet, \text{ push in } \mu'(p) \text{ the element } 0, \ Post(p,t) \text{ times.} \end{cases}$$

### *B. Calculus of the state zone graph*

The verification of time Petri nets properties is based on abstraction, whose aim is to represent, by removing some irrelevant details, the infinite state space of the model by a finite graph, which preserves properties of interest. There are two well known abstraction techniques used in the literature for the reachability analysis of time Petri nets [4], [8], [12]: the *state class graph* [3] and the *zone based graph* [7]. The basic difference between the two abstractions is the state definition. In the state class graph method, states are defined by a marking and a function, which associates a firing interval with each transition. In the state zone graph, states are defined using clocks as explained in Section II. Both methods can be used in the context of our semantics in a similar way as done for other semantics.

We have implemented a preliminary version of the construction procedure of the zone based graph [4], w.r.t. our semantics. As an example, we report in Fig. 6, the zone based graph obtained for the TPN at Fig. 5.

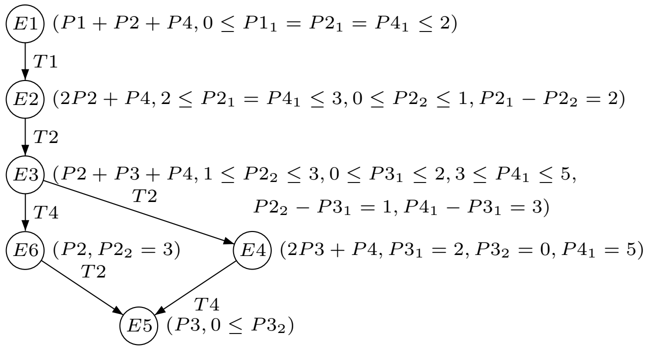


Fig. 6. The state zone w.r.t. our semantics of the TPN at Fig. 5

## V. CONCLUSION

In this paper, we have studied time Petri nets model and its different semantics: the intermediate, the atomic and the persistent atomic one. We have shown that the intermediate semantics is not an age semantics, and wrongly used can involve some unexpected and incoherent behaviours: some tokens may be used by a transition even if, they did not wait the required time. We observ the same phenomenons in the context of the atomic and persistent atomic semantics.

To cope with this problem, we have proposed a semantics based on the age of tokens to deal with conflicts. This semantics associates clocks with tokens and then allows to handle appropriately conflicts, respecting the waiting time of tokens.

We have shown, by means of examples, that a TPN may be bounded w.r.t. the intermediate, atomic or persistent atomic semantics but unbounded w.r.t. our semantics. Reciprocally, a TPN may be unbounded w.r.t. the intermediate, atomic or persistent atomic semantics but bounded w.r.t. our semantics. Therefore, there is no relationship between properties of the TPN w.r.t. our semantics and others.

Note that in non-conflict situation, our semantics can simulate the intermediate semantics by adding a self-loop place to each transition.

Finally, as a perspective, we will investigate the comparison of the expressiveness of our semantics with the others.

## REFERENCES


[1] T. Aura and J. Lilius, *A causal semantics for time Petri nets*, Theoretical Computer Science 243(2), pages 409-447, 2000.

[2] B. Bérard, F. Cassez, S. Haddad, D. Lime, Didier and O. H. Roux, *Comparison of different semantics for time Petri nets*, Third International Symposium on Automated technology for verification and analysis, volume 3707 of LNCS, pages 293-307, 2005.

[3] B. Berthomieu and M. Diaz, *Modeling and verification of time dependent systems using time Petri nets*, IEEE Transactions on Software Engineering, volume 17, number 3, 1991.

[4] H. Boucheneb, G. Gardey, and O. (H.) Roux, *TCTL model checking of time Petri nets*, Journal of Logic and Computation, volume 19(6), pages 1509-1540, December 2009.

[5] M. Boyer and M. Diaz, *Multiple Enabledness of Transitions in Petri Nets with Time*, PNPM01, 2001

[6] M. Boyer and O. H. Roux, *On the compared expressiveness of arc, place and transition time Petri Nets*, journal Fundamenta Informaticae, volume 88, $n^o$ 3, pages 225-249, 2008.


[2]The operation $\mu + d$ increments with $d$ time units all clock values within queues of $\mu$.

[7] G. Gardey, O. H. Roux and O. F. Roux, *Using zone graph method for computing the state space of a Time Petri Net*, Springer-Verlag, volume 2791 of LNCS, 2004.
[8] R. Hadjidj and H. Boucheneb, *Improving state class constructions for CTL* model checking of Time Petri Nets*, International Journal on Software Tools Technology Transfer (STTT), volume 10, number 2, pages 167-184, 2008.
[9] W. Khansa, J.-P Denat and S. Collart-Dutilleul, *P-Time Petri Nets for manufacturing systems*, International Workshop on Discrete Event Systems, WODES, pages 94-102, 1996.
[10] P. Merlin, *A study of the recoverability of computer system*, PhD Thesis, University of California, 1974.
[11] C. J. Myers, T. G. Rokicki, T. H.-Y. Meng, *POSET timing and its application to the timed cirucits*, in IEEE Transactions on CAD, 18(6), 1999.
[12] W. Penczek and A. Pólrola, *Specification and Model Checking of Temporal Properties in Time Petri Nets and Timed Automata*, In Proc. of ICATPN, Springer-Verlag, volume 3099 of LNCS, pages 37-76, Springer–Verlag, 2004.
[13] C. Ramchandani, *Analysis of asynchronous concurrent systems by timed Petri nets*, PhD Thesis, Massachusetts Institute of Technology, 1974.
[14] B. Walter, *Timed net for modeling and analysing protocols with time*, IFIP Conference on Protocol Specification Testing and Verification, 1983.