Time Petri Nets with Action Duration: A True Concurrency Real-Time Model

Nabil Belala, Djamel Eddine Saidouni, Radja Boukharrou, Ahmed Chawki Chaouche, Adel Seraoui, Asma Chachoua

TIME PETRI NETS WITH ACTION DURATION: A TRUE CONCURRENCY REAL TIME MODEL

D-E. Saïdouni, N. Belala, R. Boukharrou, A- C Chaouche, A. Seraoui, and A. Chachoua

MISC Laboratory, Constantine II, Algeria

ABSTRACT

The design of real-time systems needs a high-level specification model supporting at the same time timing constraints and actions duration. We introduce in this paper an extension of Petri Nets called Time Petri Nets with Action Duration (DTPN) where time is associated with transitions. In DTPN, the firing of transitions is bound to a time interval and transitions represent actions which have explicit durations. We give an operational semantics for DTPN in terms of Durational Action Timed Automata (DATA). DTPN considers both timing constraints and durations under a true-concurrency semantics with an aim of better expressing concurrent and parallel behaviours of real-time systems.

Keywords: *Real-time systems, DTPN, maximality semantics, Actions duration, Durational Action Timed Automata*

INTRODUCTION

Petri Nets are a well-established formal model for the specification of distributed and concurrent systems. This model is very attractive by its ability to capture causal and parallel behaviours of these systems. Since its introduction, timed models based on Petri nets have been extensively studied for the specification and verification of real-time systems.

The two main extensions of Petri Nets with time are Timed Petri Nets (TdPN) and Time Petri Nets (TPN) (Ramchandani, 1974; Merlin, 1974). In TdPNs, delays were first associated with transitions (T-TdPN) and then to places (P-TdPN) (Ramchandani, 1974; Sifakis, 1977). The two corresponding subclasses namely T-TdPN and P-TdPN are expressively equivalent (Ramchandani, 1974; Sifakis, 1977) and are a subclass of TPNs. Thus, a time delay can represent a minimum duration of firing or a minimum residence time of a token in a place. Informally, the TdPN uses the notion of duration as opposed to the notion of period of TPN.

In TPNs, temporal extension is expressed as an interval associated mainly to transitions (T-TPNs), to places (P-TPNs) or arcs (A-TPNs) (Merlin, 1974; Khansa, 1997; Walter, 1983). Regarding the expressiveness of T-TPN and P-TPN, Khansa et al. (1996) showed that these two models are incomparable. A-TPNs and P-TPNs are similar; however, the only difference concerns the strong semantics of P-TPNs and the lazy semantics of A-TPNs (Boyer, 1997). At last, T-TPNs form a subclass of Time Stream Petri Nets (Emerson, 1990) that have been introduced to model multimedia applications.

TPNs are primarily used for performance analysis. In these models the firing of transitions is of null duration. They natively express specifications in time. In explaining beginnings and ends of actions with specification of time progress, they can also express specifications in duration. However, this manner of modelling the action durations has many disadvantages. First, the size of the associated semantic structure is increased. This phenomenon is known as the state space combinatorial explosion problem. Second, the obtained specification structurally keeps out the statement of system to be specified. Third, the

underlying semantics, usually the interleaving semantics, supposes structural and temporal atomicity of actions, i.e., actions are indivisible and have null duration. Moreover, this semantics gives abstracts to the parallel execution of actions.

With the assumption that the firing of each transition corresponds to the execution of a divisible action with duration, our goal is to exploit a model which permits expressing true-concurrency in a natural way without splitting actions into their start and end events. To do this, we propose at first an extension of TPN model called Time Petri Nets with Duration Action (DTPN). In this model, two annotations are associated to each transition, namely its timing constraint that restricts the date at which it can be fired and the duration of its corresponding action. Consequently, DTPNs can be considered as a generalization of Merlin's TPNs, T-TdPNs and P-TdPN (Merlin, 1974; Ramchandani, 1974; Sifakis, 1977). Then, we give true-concurrency semantics to DTPN in terms of maximality semantics (Devillers, 1992; Courtiat & Saïdouni, 1995; Saïdouni & Courtiat, 2003; Saïdouni Belala & Bouneb, 2008; Saïdouni, Belala & Bouneb, 2009). This semantics has been proven necessary and sufficient for the action refinement and for action durations (Saïdouni, 1996). The underlying model is a Durational Action Timed Automaton (DATA) (Saïdouni & Belala, 2006).

DATA provides an abstract model for real-time systems based on true concurrency semantics. It is very near syntactically to Timed Automata (Alur & Dill, 1994) on which transitions are specified in terms of two related conditions (guard and deadline) expressing respectively, possibility and forcing of execution by stopping time progress. As a main result, DATA allows the verification of new properties related to simultaneous progress of actions at different states of the system.

The paper is organized as follows. First, we introduce the Time Petri Nets with Action Duration model. Next, we recall Durational Action Timed Automata with its semantics. In the follow section, we show the operational construction of DATA associated to DTPN, illustrative examples and DATA generation algorithm. Then, a case study of multimedia document modelled with DTPN is presented. After, we discuss the advantages of the proposed model. Finally, we give some conclusions and perspectives of our work.

TIME PETRI NETS WITH ACTION DURATION

Syntax

The basic idea of Time Petri Nets with Action Duration (DTPN) is to associate two dates *min* and *max* with each transition that define its firing interval (temporal interval). Although the firing of a transition is instantaneous, the execution duration of the action associated to this transition may have non-null duration. For example, let t be a transition associated to the action which has a duration d. If θ is the enabling date of t then the firing of t will be in the time interval $[\theta + min, \theta + max]$. The firing of t marks the start of execution of the associated action.

A place of a DTPN corresponds to two sets: a set of *available tokens* or *free tokens* and a set of *unavailable tokens* or *bound tokens*. Unavailable tokens, put on the right side of a place, are bound to the firing of transitions associated to actions that are currently running. In a DTPN, an unavailable token becomes available if the end of execution of the action associated to the transition that produced this token is reached. A token in place p at the time ϑ becomes *available* (in the left side of p) at the time $\vartheta + d$. Thus, the token is bound to the firing of the transition during the interval $[\vartheta, \vartheta + d[$ and it becomes free at the time $\vartheta + d$.

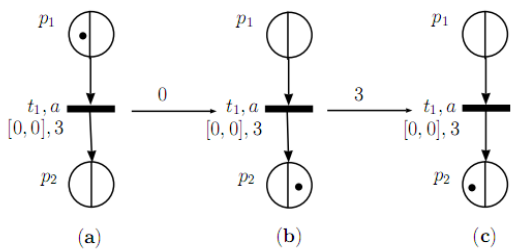


Figure 1. Marked DTPN

In Figure 1.(a), the token in place $p1$ is not bound to any transition. This token is called *free*. In the case when the transition would be fired, it could be argued that the action associated to the firing of $t1$ has started its execution. This is marked by the presence of the token in place $p2$ (Figure 1.(b)). Thus, the token in place $p2$ is bound to the firing of t_1, but after completion of the action a, i.e. after 3 units of time, this token will become free (Figure 1.(c)). In a place, the set of free tokens will be denoted by FT , while bound tokens set will be denoted by BT.

Definition 1 (DTPN): Let $\mathbb{T}$ be a non-negative temporal domain (like $\mathbb{Q}+$ or $\mathbb{R}+$) and Act be a finite set of actions, i.e. an alphabet[i]. A Time Petri Net with Action Duration (DTPN) on $\mathbb{T}$ and of support Act is a tuple $N = (P, T, B, F, \lambda, SI, \Gamma)$ such that

- $Q = (P, T, F, B)$ is a Petri net where P is a set of places, T is a set of transitions [ii] such that $P \cap T = \emptyset$. $B : P \times T \to \mathbb{N}$ is a backward incidence function such that $B(pi, tj)$ represents the arc weight from pi to tj and $F : P \times T \to \mathbb{N}$ is a forward incidence function such that $F(pi, tj)$ represents arc weight from tj to pi,
- $\lambda : T \to Act \cup \{\tau\}$ is a labelling function of a DTPN. If $\lambda(t) \in Act$ then t is called observable or external,
- $SI : T \to \mathbb{T} \times \mathbb{T} \cup \infty$ is a function that associates to each transition a static firing interval,
- $\Gamma : Act \to D$[iii] is a function that associates to each action its static duration.

I is the set of all intervals of a DTPN such as $I(t) = [min, max]$ is the interval associated to the transition t . We denote by $\downarrow I(t) = min$ and $\uparrow I(t) = max$ two functions which give respectively the lower and upper bound of an interval.

As commonly in use in the literature, we write °t (resp. $t°$) to denote the set of places such that $°t = \{p \in P/B(p,t) > 0\}$ (resp. $t° = \{p \in P/F(p,t) > 0\}$), and °$p$ (resp. $p°$) to represent the set of transitions such that $°p = \{t \in T/F(p,t) > 0\}$ (resp. $p° = \{t \in T/B(p,t) > 0\}$).

Semantics

Explicit Actions Durations. A real-valued clock is associated with each token. This token is called bound when the duration condition associated with it, in the form of $x \geq d$ where x is the associated clock and d is the static action duration, is not satisfied. The token becomes free as soon as this duration condition will be satisfied. For example, let us consider DTPN of Figure 2.(a). After the firing of the transition t_1 (Figure 2.(b)), the token remains bound to the latter for *3* units of time, so, the set of the bound tokens in p_2 is $BT = \{t_{1_{x \geq 3}}\}$. This token will become free at the moment of the satisfaction of the duration condition $x \geq 3$.

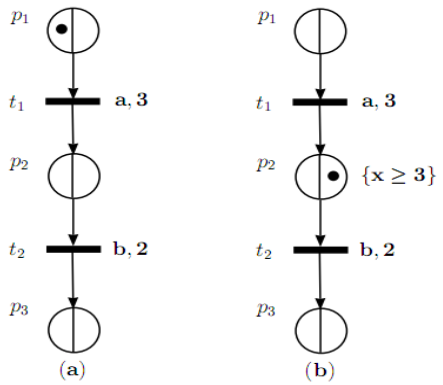


Figure 2. Petri net with explicit actions durations.

***Definition2* (Bound token)**: A bound token is an element from $T \times \mathcal{H} \times D$ noted by (t, x, d) (or $(t_{F_{x(d)}})$) such that

- t is the producing transition of this token,
- x is the clock associated to the start of the action associated to t,
- d is the duration of this action,
- $F_{x(d)} = \{x \geq d\}$ is the ending condition of the action associated to t ($F_{x(d)}$ is called also duration condition).

A question that arises concerns tokens which are bound to the same transition. To see that, consider the Petri net of Figure 3.(a). With the firing of the transition t_1, we obtain the derivation of Figure 3.(b). The right side BT of the place p_2 contains two tokens bound to the firing $t_{1_{x\geq 3}}$, i.e. $BT = \{t_{1_{x\geq 3}}, t_{1_{x\geq 3}}\}$. Since BT is a set, we consider that bound tokens denoted with the same clock are defined as a tuple (n, t, x, d) of $\mathbb{N} \times T \times \mathcal{H} \times D$, also denoted $nt_{F_{x(d)}}$, where n is the number of instances. We denote by $BT = \{n_1 t_{1_{x_1 \geq d_1}}, \ldots, n_m t_{m_{x_m \geq d_m}}\}$ the set (possibly empty) of bound tokens. In the previous example, $BT = \{2t_{1_{x\geq 3}}\}$.

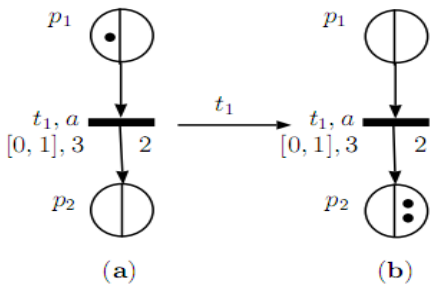


Figure 3. Petri net with an output arc of a weight greater than 1

Definition 3 (Marking of DTPN): Let $N = (P, T, B, F, \lambda, SI, \Gamma)$ be a DTPN, a marking of N is a function $M : P \rightarrow 2^{\mathbb{N}\times T\cup\{\delta\}\times\mathcal{H}\cup\{C\emptyset\}\times D} \times 2^{\mathbb{N}\times T\times H\times D}$. Among others, the marking $M(p)$ of a place $p \in P$ is a pair (FT, BT) such that $FT \in 2^{\mathbb{N}\times T\cup\{\delta\}\times H\cup\{C\emptyset\}\times D}$ and $BT \in 2^{\mathbb{N}\times T\times H\times D}$ denote respectively the set (possibly empty) of free tokens and the set (possibly empty) of bound tokens in the place p.

In what follows, a DTPN with a marking is called *configuration* which denoted a *state*. $|M(p)|$ denotes the total number of tokens in a place p. If $M(p) = (FT, BT)$ such that $FT = \{n_1 t_{1_{x_1\geq d_1}}, \ldots, n_k t_{k_{x_k\geq d_k}}\}$ and $BT = \{n'_1 t'_{1_{x'_1\geq d'_1}}, \ldots, n'_m t'_{m_{x'_m\geq d'_m}}\}$ then $|M(p)| = |FT| + |BT|$ with $|FT| = \sum_{i=1}^{k} n_i$ and $|BT| = \sum_{i=1}^{m} n'_i$.

Explicit Timing Constraints. In DTPN's, a transition t can be fired if its clock value is in the associated firing interval. Therefore, the firing of the transition is conditioned by the satisfaction of the guard $\downarrow I(t) \leq x \leq \uparrow I(t)$. For example, let us observe the DTPN of Figure 4. In the initial configuration, all the tokens are available and the duration conditions are all satisfied. They are defined by $C_\emptyset \geq 0$ where $C_\emptyset$ denotes a particular clock created and initialized at the enabling time of the system and which is associated to a particular transition δ launched at the system start-up. While the firing of the transition t_1 will be in the interval $[2, 3]$, the switch of the configuration shown in Figure 4 is conditioned by the satisfaction of the guard $2 \leq C_\emptyset \leq 3$.

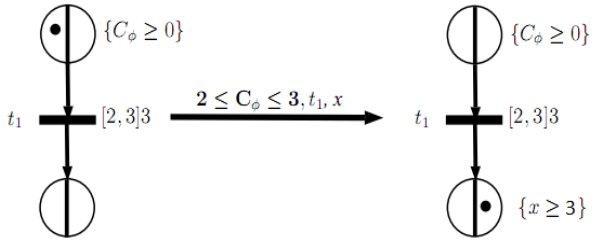


Figure 4. Firing of t1

Consider now the example of the DTPN of Figure 5. If we consider that one token in the place p_2 is bound to the firing of t_1 ($t_{1_{x\geq 2}}$) and the other one is bound to the firing of t_2 ($t_{2_{y\geq 3}}$) then we can fire the transition t_3 using the token identified by the clock x (resp. y) if and only if the condition $x \geq 2$ (resp. $y \geq 3$) is satisfied but in the relative interval $[2, 3]$.

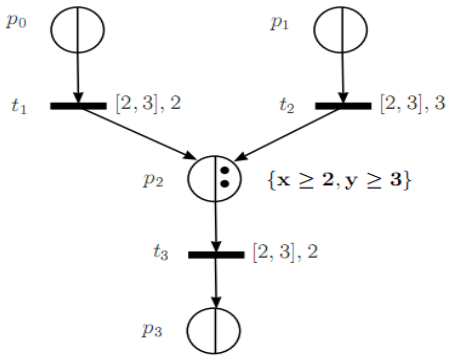


Figure 5. Identification of consumed tokens.

Therefore, the firing of t_3 is causally depending on the clock x (resp. y) if and only if both the duration condition and the guard $2 + 2 \leq x \leq 3 + 2$ (resp. $3 + 2 \leq y \leq 3 + 3$) are satisfied. By a first firing of t_3, we obtain the configuration C_1 (resp. C_2) of Figure 6 and through a second firing of the same transition, we obtain the configuration C_3. The multienabling of a transition expresses the auto-concurrency, so, starting from C_1 and C_2 one can reach C_3.

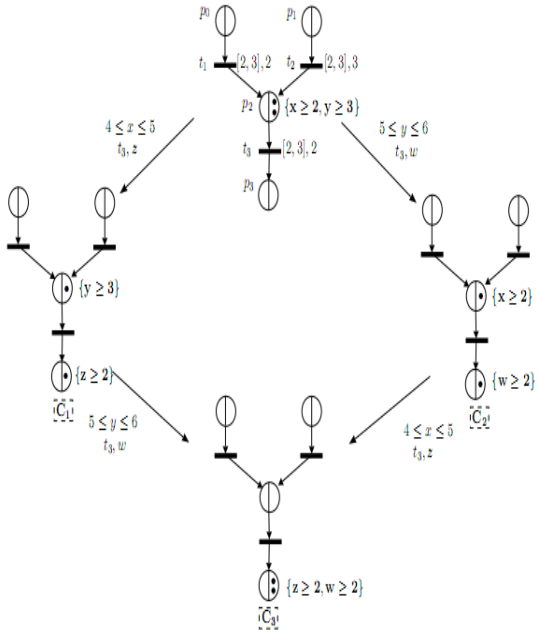


Figure 6. Clocks identifying consumed tokens.

DURATIONAL ACTION TIMED AUTOMATA

The DATA model (Saïdouni & Belala, 2006; Belala, 2010) is a timed model defined by a timed transition system over an alphabet representing actions to be executed. Structural and temporal non-atomicity of actions are supported by DATA. i.e., actions may be divisible and of non-null durations.

The DATA model supports the notions of urgency and deadlines as timing constraints of the system. An action duration is expressed by a duration condition associated to the states of the model. On the other hand, timing constraints due to restrictions on the enabling domain of an action are expressed by the *enabling constraint* G (for *guard*) and by *urgency constraint* D (for *deadline*) at the level of DATA transitions. In addition, a transition represents only the start of an action, end of execution is captured by the corresponding duration. On the target state of a transition, a timed expression manifests that the action is potentially in execution.

From operational point of view, with each action is associated a *clock* which is *reset* at the start of the action. This clock will be used in the construction of the timing constraints as guards of the transitions. This model is illustrated by the example of Figure 7 composed of two states and a transition labelled with an action a of duration 2 units of time.

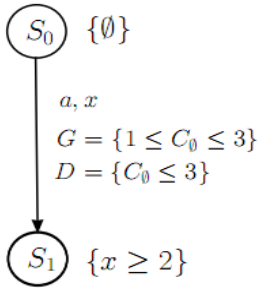


Figure 7. Example of DATA.

From the initial state S_0 of the illustrative DATA, the execution of the action a leads to a reset of the clock x associated with it.

The expression $x \geq 2$ in state S_1 represents a duration condition on the action a and means that a is potentially in execution until the clock x reaches the value 2. The action a does not wait for the end of any other action, so the clock designated by the enabling domain of this action will be $C_\emptyset$. This enabling domain will be expressed by the guard and the deadline on the clock $C_\emptyset$ ($1 \leq C_\emptyset \leq 3$).

Formalization

Definition 4 : Let $\mathcal{H}$ be a set of clocks with non-negative values (within a time domain $\mathbb{T}$, like $\mathbb{Q}^+$ or $\mathbb{R}^+$). The set $\Phi_t(\mathcal{H})$ of temporal constraints γ over $\mathcal{H}$ is defined by $\gamma ::= x \sim t$ where x is a clock in $\mathcal{H}$, $\sim \in \{=, <, >, \leq, \geq\}$ and $t \in T$. F_x will be used to indicate a constraint of the form x ~ t. A valuation (or interpretation) v for H is a function which associates to each $x \in \mathcal{H}$ a value in $\mathbb{T}$. A valuation v for $\mathcal{H}$ satisfies a temporal constraint γ over $\mathcal{H}$ iff γ is true by using clock values given by v. For $I \subseteq \mathcal{H}$, $[I \rightarrow 0]v$ indicates the valuation for $\mathcal{H}$ which assigns value 0 to each $x \in I$, and agrees with v over the other clocks of $\mathcal{H}$. The set of all valuations for H is noted $\Xi(\mathcal{H})$. The satisfaction relation $\vDash$ for temporal constraints is defined over the set of valuations for $\mathcal{H}$ by : $v \vDash x \sim t \Leftrightarrow v(x) \sim t$ such that $v \in \Xi(\mathcal{H})$. $2^{\mathbb{T}}_{fn}$ is used to denote the set of finite subsets of a set $\mathbb{T}$.

Definition 5 : A DATA A is a tuple $(S, L_s, s_0, \mathcal{H}, T_D)$ of the support Act where

- S is a finite set of states,
- $L_s : S \rightarrow 2^{\Phi_t(H)}_{\text{fn}}$ is a function which assigns to each state s the set F of ending conditions (duration conditions) of actions possibly in execution in s,
- $s_0 \in S$ is the initial state, such that $L_s(s_0) = \emptyset$,
- $\mathcal{H}$ is a finite set of clocks,
- $T_D \subseteq S \times 2^{\Phi_t(H)}_{\text{fn}} \times 2^{\Phi_t(H)}_{\text{fn}} \times Act \times \mathcal{H} \times S$ is the set of transitions.

A transition (s, G, D, a, x, s') represents a switch from state s to state s' by starting execution of action a and resetting clock x. G is the corresponding guard which must be satisfied to fire this transition. D is the corresponding deadline which requires, at the moment of its satisfaction, that action a must occur. (s, G, D, a, x, s') can be written $s \xrightarrow{G,D,a,x} s'$.

Definition 6 : The semantics of a DATA $A = (S, L_s, s_0, H, T_D)$ is defined by associating to it an infinite transitions system S_A over $Act \cup \mathbb{R}^+$. A *state* of S_A (or *configuration*) is a pair $\langle s, v \rangle$ such that s is a state of A and v is a valuation for $\mathcal{H}$. A configuration $\langle s_0, v_0 \rangle$ is initial if s_0 is the initial state of A and $\forall x \in \mathcal{H}$, $v_0(x) = 0$. Two types of transitions between S_A configurations are possible, and which correspond respectively to time passing (rules $RA1$ and $RA2$) and the launching of a transition from A (rule RD).

$$\text{(RA1)}\ \frac{d \in \mathbb{R}^+\ \forall d \leq d,\ v + d' \nvDash \mathfrak{D}}{\langle s, v \rangle \xrightarrow{d} \langle s, v + d \rangle}$$

$$\text{(RA2)}\ \frac{\varepsilon \in \mathbb{R}^+\ v + \varepsilon \vDash \mathfrak{D} \wedge \varepsilon \in \eta}{\langle s, v \rangle \xrightarrow{\varepsilon} \langle s, v + \varepsilon \rangle}$$

$$\text{(RD)}\ \frac{(s, G, D, a, x, s') \in T_D\ \ v \vDash G}{\langle s, v \rangle \xrightarrow{a} \langle s', [\{x\} \rightarrow 0]v \rangle}$$

Where η is the smallest real quantity of time in which no action occurs (Belala, 2010). In RA rules, $\mathfrak{D} = \bigvee_{i \in I} D_i$ where $\{ (s, G_i, D_i, a_i, x_i, s_i) \}_{i \in I}$ is the set of all transitions stemming from state s. Indeed, whenever a D_i holds, time cannot progress regardless of the other D_i.

Note that if one wants to guarantee that at least a transition could be drawn starting from a state if time cannot progress any more within this state, one requires that the formula $D_i \Rightarrow G_i$ be satisfied.

Remark: For urgency domains, we require that deadline can be only of the form "$x \leq t$" or "$x < t$".

OPERATIONAL CONSTRUCTION OF DATA ASSOCIATED TO DTPN

In the following paragraph, we give some preliminary definitions that enable us to propose a generation method of a temporal marking graph in the context of the maximality semantics.

Preliminary Definitions

***Definition 7*:** Let $N = (P, T, B, F, \lambda, SI, \Gamma)$ be a DTPN with a marking M and a transition t :

- The set of ending conditions of actions potentially in execution in M is the set of all conditions on clocks identifying bound tokens in the marking M . Formally, the function L_M will be used to calculate this set. L_M can be defined as
  - $LM(M) = \bigcup_{p \in P} \bigcup_{i=1}^{m} F_{x_i}(d_i)$ with $F_{x_i}(d_i) \in BT$ such that $M(p) = (FT, BT)$ and $BT = \{(n_1, t_1, F_{x_1(d_1)}), \ldots, (n_m, t_m, F_{x_m}(d_m))\}$.
- The set of all ending conditions of actions in M is calculated by the function ψ which is defined as
  - $\psi(M) = \bigcup_{p \in P} \psi_p(M)$ such that $\psi_p(M) = \bigcup_{i=1}^{n} F_{xi}(d_i)$ with $x_i \in \mathcal{H}$ and $F_{x_i}(d_i) \in FT \cup BT$ of the place p $(M(p) = (FT, BT))$.
- Let $h = \{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{H}$ be a finite non-empty set of clocks. To make free the tokens bound to clocks x_1, x_2, ..., x_n we define recursively the function $\mathfrak{M}_{\mathfrak{F}}(h, M)$ as
  - $\mathfrak{M}^1_{\mathfrak{F}}(\{x_1\}, M) = \mathfrak{M}_{\mathfrak{F}}(x_1, M)$
  - $\mathfrak{M}^n_{\mathfrak{F}}(\{x_1, x_2, \ldots, x_n\}, M) = \mathfrak{M}^{n-1}_{\mathfrak{F}}(\{x_1, \ldots, x_{n-1}\}, \mathfrak{M}_{\mathfrak{F}}(x_n, M))$
  - $\mathfrak{M}_{\mathfrak{F}}(x, M) = M'$ such that for all $p \in P$, if $M(p) = (FT, BT)$ then
    - If there is $(n, t, F_x(d)) \in BT$ then $M'(p) = (FT \cup \{(n, t, F_x(d))\}, BT - \{(n, t, F_x(d))\})$[iv]
    - If $(n, t, F_x(d)) \in FT$ and $n = 0$ then $M'(p) = (FT - \{(n, t, F_x(d))\}, BT)$
    - $M'(p) = M(p)$, otherwise.
- The transition t is said to be fired by the marking M iff t is enabled by this marking and it is not excluded by another transition enabled by M i.e., its upper bound of its firing interval is smaller than the lower bounds of all other conflicted transitions. Formally, t is fired by the marking M iff $\forall p \in {}^\circ t\, |M(p)| \geq B(p, t)$ and $\{\nexists t' \in p^\circ / \forall p' \in {}^\circ t' |M(p')| \geq$ B(p', t') and $\uparrow$I(t')<$\downarrow$I(t)$\}$. The set of all transitions enabled by the marking M is noted $enabled(M)$.

Let us consider the example of Figure 8 in which t_2 and t_3 are two transitions enabled by the same marking. However, by applying the rule of firing, only the transition t_2 will be fired. i.e, $enabled(M) = \{t2\}$.

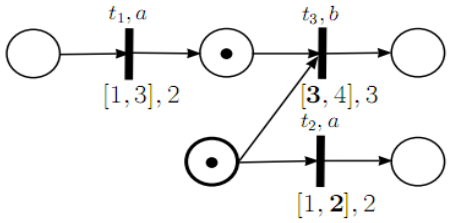


Figure 8. Example of two conflicted transitions.

- $get : 2^E - \{\emptyset\} \rightarrow E$ is a function which selects an element of its operand[v], i.e., it satisfies $get(X) \in X$ for any $X \in 2^E - \{\emptyset\}$.

- The guard or timing constraints for the firing of transition t by the marking M can be defined as $guard(M,t) =$

$$\begin{cases} \{\downarrow \mathrm{I(t)} \le C_{\emptyset} \le \uparrow \mathrm{I(t)}\}, & \boldsymbol{if} \left|\bigcup_{p \in {}^{\circ}t} \boldsymbol{\psi_p(M)}\right| = \mathbf{0}; \\ \{\downarrow \mathrm{I(t)} + \mathrm{d_i} \le x_i \le \uparrow \mathrm{I(t)} + \mathrm{d_i}\}, & \boldsymbol{if} \left|\bigcup_{p \in {}^{\circ}t} \boldsymbol{\psi_p(M)}\right| = \mathbf{1}; \\ \bigcup_{i=1}^{m} \{x_i \ge \downarrow \mathrm{I(t)} + \mathrm{d_i}\}, \textit{such that} & \boldsymbol{if\ m} > 1. \\ m = \left|\bigcup_{p \in {}^{\circ}t} \psi_p(M)\right| \quad \textit{and } \exists\, F_{xi}\,(d_i) \in \bigcup_{p \in {}^{\circ}t} \psi_p(M), & \end{cases}$$

- Let G be a set of timing constraints and $c_{\varkappa}$ be the smallest number c, $get(Min(G)) = \varkappa$ is a function which gives the clock with the smallest bound in G such that $\varkappa$ is compared with c in some clock constraint appearing in G.
- The function $bound(x, G)$ returns the lower bound of the clock x in the set of guards G. For example if $G = \{x \ge 5\}$ then 5 is the *bound* of x.
- The deadline or urgency constraint can be defined as $deadline(G,t) =$

$$\begin{cases} \{x \le \uparrow \mathrm{I(t)} + \mathrm{bound}(\varkappa, G) - \downarrow \mathrm{I(t)}\}, & \boldsymbol{if} |\boldsymbol{G}| > 1; \\ \{x_i \le \max_i\} \ \textit{with } G = \{min_i \le x_i \le max_i\} & \boldsymbol{if} |\boldsymbol{G}| = \mathbf{1}; \end{cases}$$

- The marking M is said to be minimal for the firing of a transition t iff $|M(p)| = B(p,t)$ for all $p \in P$.
- Let M_1 and M_2 be two markings of N. $M_1 \Subset M_2$ iff $\forall p \in P$, if $M_1(p) = (FT_1, BT_1)$ and $M_2(p) = (FT_2, BT_2)$ then $FT_1 \Subset FT_2$ and $BT_1 \Subset BT_2$ such that the relation $\Subset$ is extended to bound tokens and free tokens sets. $BT_1 \Subset BT_2$ iff $\forall (n_1, t, F_x(d)) \in BT_1$, $\exists (n_2, t, F_x(d)) \in BT_2$ such that $n_1 \le n_2$. $FT_1 \Subset FT_2$ iff $\forall (n_1, t, F_x(d)) \in FT_1$, $\exists (n_2, t, F_x(d)) \in FT_2$ such that $n1 \le n2$.
- If $M_1 \Subset M_2$ the difference $M_2 - M_1$ is a marking M_3 such that for all $p \in P$, if $M_1(p) = (FT_1, BT_1)$ and $M_2(p) = (FT_2, BT_2)$ then $M_3(p) = (FT_3, BT_3)$ with
  - $\forall (n_1, t, F_x(d)) \in FT_1$, $(n_2, t, F_x(d)) \in FT_2$, if $n_1 \ne n_2$ then $(n_2 - n_1, t, F_x(d)) \in FT_3$.
  - $\forall (n_1, t, F_x(d)) \in BT_1$, $(n_2, t, F_x(d)) \in BT_2$, if $n_1 \ne n_2$ then $(n_2 - n_1, t, F_x(d)) \in BT_3$.
- $Min(M,t) = \{M'/M' \Subset M\}$ and M' is minimal for the firing of t.
- $occur(t, x, M) = M'$ is a function applied with parameters t, $x \notin L_M(M)$ and M. It generates a new marking M' from M following the firing of transition t where the associated clock will be x. Formally, $\forall p \in P$, if $M(p) = (FT, BT)$ then $M'(p) = (FT, BT')$ with $BT' = BT \cup \{F(p,t), t, F_x(d)\}$ $/d = \Gamma(\lambda(t))$ iff $p \in t^{\circ}$ and $BT' = BT$ otherwise. Hence, M' is the resulting marking from the addition of tokens bound to t to the marking M.

Temporal Marking Graph Construction

Let $N = (P, T, B, F, M0, \lambda, SI, \Gamma)$ be a marked and a bounded DTPN with $\forall t \in T$, ${}^{\circ}t \ne \emptyset$ and $t^{\circ} \ne \emptyset$. The *Temporal Marking Graph* (TMG) labelled by λ associated to N is a graph in which the states are defined by all reachable markings from the initial marking M_0 and the transitions between states are labelled according to the derivation rule of Definition 8.

Definition 8 **:** Let M be a reachable marking of the DTPN $N = (P, T, B, F, M0, \lambda, SI, \Gamma)$. If $t \in enabled(M)$ then for all $M'' \in Min(M,t)$, $G = guard(M'', t)$, $D = deadline(G,t)$, $a = \lambda(t)$, $E = \bigcup_{F_x(d) \in \psi(M'')} x$ and $\mathrm{M'''} = \mathfrak{M}_{\mathfrak{F}}(E, \mathrm{M} - \mathrm{M''})$, the following derivation is possible: $M \xrightarrow{G,D,a,x} M'$ such that

- E is the set of clocks associated with actions in which the end is required for the launch of the action related to the firing of t,
- x= $get(H - \psi(M'''))$,
- $M' = occur(t, x, M''')$,
- G is the set of guards,
- D is the deadline.

Note that the bound clocks may not be selected. However, a clock which becomes free can be reused.

Property

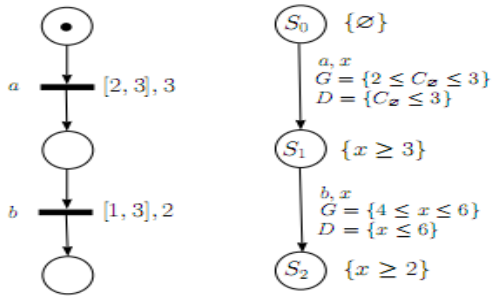
Proposition1. Let $N = (P, T, B, F, M0, \lambda, SI, \Gamma)$ be a marked DTPN and TMG its finite Temporal Marking Graph built according to Definition 8. The structure $(\mathcal{M}, L_M, M_0, \mathcal{H}, T)$ is a DATA with

- $TMG = (\mathcal{M}, M_0, T)$ the Temporal Marking Graph associated to N such that
  - $\mathcal{M}$ is the set of states,
  - M_0 is the initial state with $L_M(M_0) = \emptyset$,
  - $T \subseteq \mathcal{M} \times 2^{\Phi_t(H)}_{\text{fn}} \times 2^{\Phi_t(H)}_{\text{fn}} \times Act \times \mathcal{H} \times \mathcal{M}$ is the set of transitions (derivations) of TMG. A derivation (M, G, D, a, x, M') represents the switch from the state M to state $'$ $(M, M' \in \mathcal{M})$, by launching the execution of the action a and using the clock x.
- $L_M : \mathcal{M} \to 2^{\Phi_t(H)}_{\text{fn}}$ is the function defined above,
- $\mathcal{H}$ is the finite set of clocks.

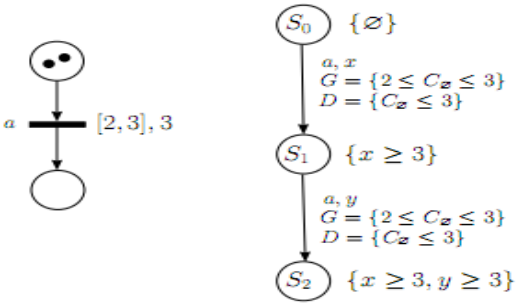
Proof. In order to show the coherence of Proposition 1, the following deductions are trivial. At first, let us observe the initial marking M_0 of the DTPN which contains only free tokens, therefore $L_M(M_0) = \emptyset$. In the other hand, we have in the initial state of DATA no action potentially in execution which implies that this initial state is exactly the initial state of TMG. Furthermore, M is the set of states defined by the set of reachable markings obtained from the initial marking M_0. Then, $L_M(M)$ gives conditions on clocks identifying only bound tokens in M which present exactly the duration conditions of actions potentially in execution in the corresponding DATA state.

Some Examples

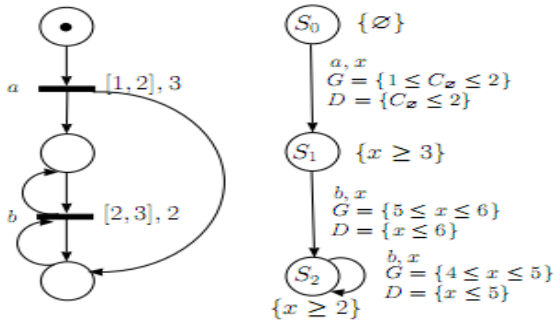
Figure 9 shows some examples of marked DTPNs with their corresponding DATA's.



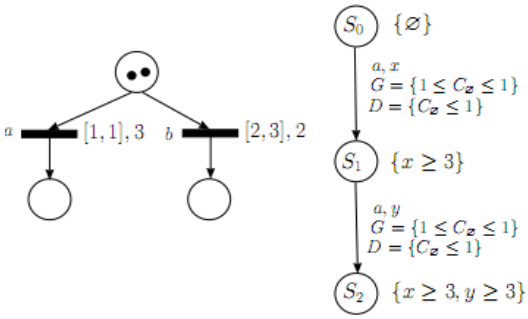
(a) Sequence



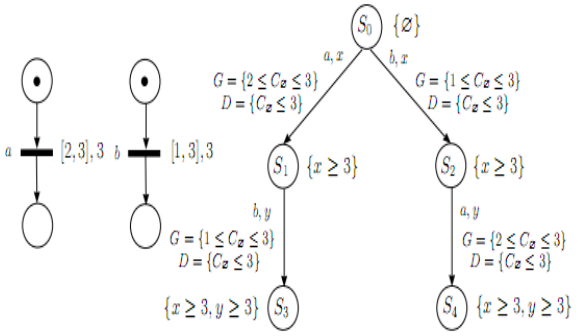
(b) Auto-concurrency



(c) Cyclic behaviour



(d) Structural conflict



(e) Parallel behaviour

Figure 9. Different examples

Generation Algorithm

Algorithm 1 gives the construction of the reachability graph of a marked DTPN. The obtained graph is a DATA. It uses a Waiting set of markings M_{NT} already created but not processed yet. From the initial marking M_0, it calculates enabled transitions and then it builds the set of successor markings M_{succs} by the firing of enabled transitions one by one. As result, we have new derivations in the form (M, G, D, a, x, M') added to the underlying graph. These markings will be added to the Waiting set M_{NT}, if they do not exist, to treat them.

Algorithm 1: DATA Generation Algorithm from DTPN
Data : a marked DTPN $(P, T, B, F, M_0, \lambda, SI, \Gamma)$;
Result : a DATA $(\mathcal{M}, L_M, M_0, \mathcal{H}, T_D)$;
Var : M_{NT} : set of untreated markings;
M_{succs}: set of successor markings;
Begin
$M_{NT} \leftarrow M_0$;
while $M_{NT} \neq \emptyset$ **do**
$M_{NT} \leftarrow M_{NT} \backslash \{M\}$;
Calculate $enabled(M)$;
for all $t \in enabled(M)$ **do**
Calculate the set $Min(M, t)$;
for all $M'' \in Min(M, t)$ **do**
$G \leftarrow guard(M'', t)$;
$D \leftarrow deadline(G, t)$;
$M''' \leftarrow \mathfrak{M}_{\tilde{\delta}}(E, M - M'')$;
$x \leftarrow get(\mathcal{H} - \psi(M'''))$;
$M' \leftarrow occur(t, x, M''')$;
if $M' \notin M$ **then**
$\mathcal{M} \leftarrow \mathcal{M} \cup \{M'\}$;
$M_{succs} \leftarrow M_{succs} \cup \{M'\}$;
end if
$t_D \leftarrow (M, G, D, t, x, M')$;
$T_D \leftarrow T_D \cup \{t_D\}$;
end for
end for
$M_{NT} \leftarrow M_{NT} \cup M_{succs}$;
end while
end

CASE STUDY

In order to illustrate the interest of DTPN for designing real-time systems with action duration, we use it for the specification of the multimedia document introduced in (Bornot, Sifakis & Tripakis, 1998).

Modelling. The building blocks of a multimedia document are media objects representing a piece of information which has to be played continuously for certain duration. As example, we consider the following document specification. DM is a document composed of two scenes, that is, two sub-documents D_1 and

D_2. D_1 is the introduction, composed of four media objects, namely, a video clip A, a sound clip B, a piece of music C and a user button D. The intention is that the video A is played in parallel with its sound B, while at the same time music is heard in the background. The user can stop the music by pressing the button. D_2 is the body of the document, composed of five media objects, namely, a still picture E followed by a video clip F and its sound clip G, which determine the presentation of an animation H and a diagram O. The duration intervals of the objects are as follows: $A: [15; 17]$, $B: [14; 16]$, $C: [9; 11]$, $D: [10; 13]$, $E: [5; 7]$, $F: [3; 6]$, $G: [4; 7]$, $H: [6; 12]$ and $O: [11; \infty)$. An interval $[d_1, d_2]$ associated to an action a means that this action takes at least d_1 duration and may elapse until d_2 duration. The specification of document DM is modelled by the PND (Petri Net with Deadlines) (Bornot, Sifakis, & Tripakis 1998) shown in Figure 10.

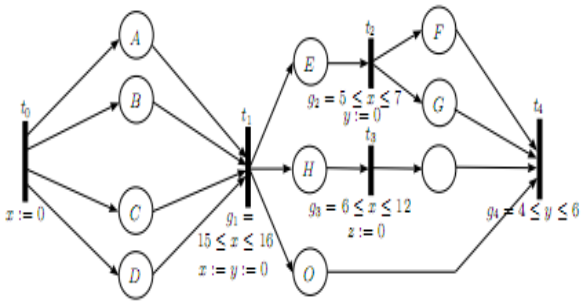


Figure 10. Example multimedia specification translated into a PND (Bornot, Sifakis & Tripakis, 1998).

The reader may remark that the transitions of PND are labelled with guards. These guards are calculated on duration intervals with operators of MADEUS language (Jourdan, Layaïda, Sabry-Ismail & Roisin, 1997). We can summarize this approach in four steps. First, the multimedia description is specified in MADEUS language. Then, the specifier calculates guards. After, this specification is modelled in PND model. Finally, the obtained PND specification is translated to a TAD (Timed Automata with Deadlines), as presented in (Bornot, Sifakis & Tripakis, 1998).

The result of specifying this example by DTPN is depicted by Figure 11. In our approach, media objects of document DM may be seen as actions having a static durations. Static duration represents the duration of transmitting a media object without any transmission problem like congestion or latency. However, the transmission may delay. The duration of this delay may be specified by a temporal interval, so, all transitions are delayable. For instance, video clip (object A) has as duration 15 units of time and may delay 2 units of time. i.e., defined by the time interval $[0, 2]$.

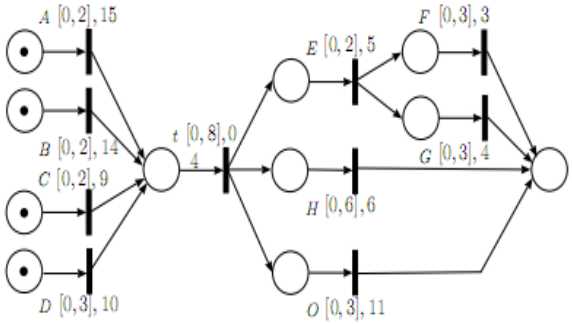


Figure 11. Example of multimedia document specification in DTPN

In the above description, the scene D_2 should be started only after the termination of scene D_1. This is conditioned by the firing of transition t (all multimedia object of D_1 are terminated). Following the same interpretation, the reader may see that the specification translates directly the multimedia example.

Verification. To investigate the verification of real-time systems, we can use algorithms and tools developed for Timed Automata (TA) (Alur & Dill, 1994) to check properties on DATA's. For instance, it is possible to check real-time properties expressed in an extension of CTL logic on bounded DTPN's (Emerson, 1990). To do this, we construct firstly the DATA structure corresponding to the DTPN specification. Then, the obtained DATA is translated to an equivalent TA which can be used with UPPAAL tool to check efficiently real-time properties (e.g. safety properties) (Bornot et al. 1998).

As example, let us check the consistency of the DTPN of Figure 11. By applying the proposed approach, a fragment of the generated DATA[vi] of this document specification is given by Figure 12. This fragment presents the sequential execution of the two sub-documents D_1 and D_2 which is modelled by the transition $(S_{15}, G, D, t, x, S_{16})$.

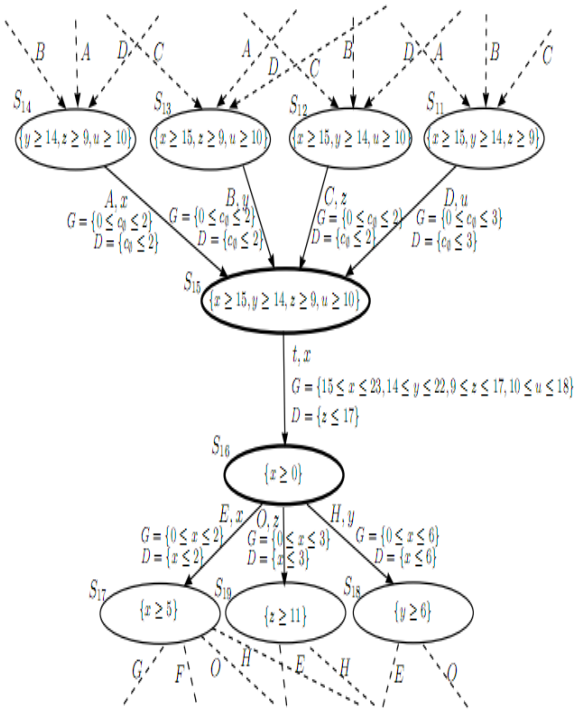


Figure 12. Fragment of DATA corresponding to multimedia document specification of Figure 11

The obtained DATA is translated to the equivalent TA of Figure 13 with some abstractions related to parallel execution of actions present in the state of the DATA. Then, using UPPAAL, we check the reachability of the TA from the initial state. The reachability is expressed by the following CTL formula ($EF\ S_{35}$). As result, the final state (S_{35}) is indeed reachable, so the consistency of the specification is verified.

Note that validation techniques can be applied directly on DATA structures using Aggregated Regions Automata (Kitouni, Hachichi, Bouaroudj & Saïdouni, 2012). That is very important for the validation based on formal methods such as model checking using tools like UPPAAL and KRONOS (Larsen, Pettersson & Yi, 1997; Guellati, Kitouni & Saïdouni, 2012; Yovine, 1997), and model based testing (Hachichi, Kitouni, Bouaroudj & Saïdouni, 2012). The use of DATA structure allows the verification of properties concerning parallel evolution of actions specified at state level of this model.

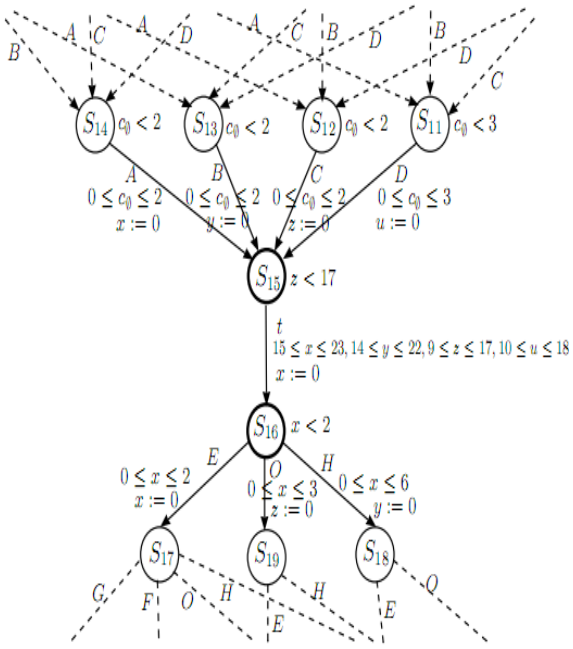


Figure 13. Fragment of TA corresponding to DATA of Figure 12

DISCUSSION AND RELATED WORK

In this work, we have given a framework to specify real-time systems. The advantages of the proposed DTPN model are the following :

Specification Advantages. DTPN gives us on one hand a natural way of modelling, it natively expresses time specification in terms of time intervals and action durations. On the other hand, it provides a simple manner of reasoning translating directly the description of real time systems.

In according to the case study, we conclude that the use of TAD supposes that the specifier is familiar with MADEUS language. Furthermore, as it may be observed in the specification, the obtained guards are not trivial to understand with respect to the system description. Consequently, the problem arises crucially during the verification result interpretation. As it is explained, DTPN overcomes all these inconveniences. Using DTPN makes multimedia documents specification easier, i.e. without a preliminary computing of guards associated to the transitions of the PND and then to its translated model TAD (Bornot, Sifakis & Tripakis, 1998).

Operational Advantages. Note that the association of two dates minimum and maximum for each transition with fixed duration of the associated action gives us an intuition that DTPN's are a native extension of T-TdPN. However, in a context of semantics that forces the firing of transitions (strong semantics[vii]), if we associate the interval $[0,+\infty[$ for any transition of T-TdPN we can see it as a DTPN. Given that timed extensions of Petri nets (T-TdPN and P-TdPN) are equivalent, DTPN's are also a generalization of P-TdPN. If all actions of a DTPN are instantaneous, this model is seen as a T-TPN. Thus, TPNs are simulated by DTPNs, where all actions have null durations. As result, we conclude that DTPNs arise as a generalization of several models which are T-TPN, P-TdPN and T-TdPN. A generalization that does not stipulate changes at the general structure of a Petri nets, the number of places (resp. the number of transitions) remains the same. In this way, DTPN is not more expressive, but, it allows us to represent with more compact manner large classes of real-time systems. This concision is often a determining factor for the verification of real-time systems due to the complexity of model checking approach which is limited by the state space combinatorial explosion.

Technical Advantages. Another advantage concerns the construction of the set of clocks. In our context, a clock is created dynamically during the generation of the marking graph. On the contrary, other models like Timed Automata and Petri Nets with Deadlines (Alur & Dill, 1994; Bornot et al., 1998) manage, at the beginning of modelling, a finite and constant number of clocks.

CONCLUSION

This paper proposes a timed extension of Petri Nets model called Time Petri Nets with Action Duration (DTPN). Two concepts have been integrated namely temporal constraints and action durations associated to transitions.

The paper gives true-concurrency semantics for DTPN's in terms of Durational Action Timed Automata (DATA). At first, we defined an operational method for generating DATA associated to DTPN specification. Then, an algorithm is proposed.

Interesting topics for future research include validation of more case studies of real-time systems in order to better benefit of the proposed model. In order to cope with the combinatorial state space explosion problem, it seems interesting to propose a distributed generation of the Durational Action Timed Automata for DTPN.

REFERENCES

Alur, R., & Dill, D.(1994). A theory of timed automata. *Theoretical Computer Science (TCS'94)*, 126, 183–235.

Belala, N.(2010). *Modèles de temps et leur intérét à la vérification formelle des systèmes temps-réel*. PhD thesis, Mentouri, Constantine, Algeria, June 2010.

Bornot, S., Sifakis, J.,& Tripakis, S. (1998). Modeling urgency in timed systems. *Lecture Notes in Computer Science (LNCS'98) :* Springer-Verlag. 1536 , 103–129.

Boyer, M. (1997). *Contribution à la modélisation des systèmes à temps constraint et application au multimedia*. PhD thesis, Universit Toulouse 3, France. Mars 1997.

Chellali, M. Z., Djoudi, B., Belala, N. & Saïdouni. (2012). From DATA to Timed Automata. *Research report, University of Mentouri, Constantine Algeria.* April 2012.

Courtiat, J. P. & Saïdouni, D. E. (1995). Relating maximality-based semantics to action refinement in process algebras. *In D. Hogrefe and S. Leue, editors, IFIP TC6/WG6.1, 7th Int. Conf. on Formal Description Techniques (FORTE'94)*. 293–308.

Devillers, R. (1992). Maximality preservation and the st-idea for action refinement. *Advances in Petri Nets*. 609, 108–151.

Diaz, M., & Senac, P. (1994). Time stream petri nets, a model for timed multimedia information. *Proc. 15th Int. Conf. Application and Theory of Petri Nets 1994, Lecture Notes in Computer Science,Springer, Berlin*. 815:219–238, June 1994.

Emerson, E. A. (1990). Temporal and modal logic. In *Elsevier, volume B, chapter 16* (pp.995–1072).

Guellati, S., Kitouni, I., & Saïdouni, D. E. (2012). Verification of durational action timed automata using uppaal. *International Journal of Computer Application (IJCA'2012)*. October 2012.

Hachichi, H., Kitouni, I., Bouaroudj, K., & Saïdouni, D. E. (2012). A graph transformation approach for testing timed systems. *The 18th International Conference on Information and Software Technologies (ICIST 2012). published as a volume of Springer-Verlag CCIS*. 319, 123–137, September 13th - 14th 2012.

Jourdan, M., Layaïda, N., Sabry-Ismail, L. & Roisin, C. (1997). Authoring and presentation environment for interactive multimedia documents. *In Proc. of the 4th Conf. on Multimedia Modelling, Singapore*. November 1997.

Khansa, W. (1997). *Réseaux de Petri P-temporels : Contribution à l'Etude des Systèmes à Evénements Discrets*. PhD thesis. Universit de Savoie, France, Mars 1997.

Khansa, W., Denat, J. P. , & Collart, D. S. (1996). P-time petri nets for manufacturing systems. *International Workshop on Discrete Event Systems, WODES'96*. pp 94–102, august 1996.

Kitouni, I., Hachichi, H., Bouaroudj, K., & Saïdouni, D. E. (2012). Durational actions timed automata: determinization and expressiveness. *International Journal of Applied Information Systems (IJAIS'2012)-ISSN : 2249-0868. 4(2)*. September 2012.

Larsen, K. G., Pettersson, P., & Yi, W. (1997) . UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer. 1(12), 134-152* . October1997.

Merlin, P. M. (1974). A study of the recoverability of computing systems. PhD thesis.

California, Irvine, CA.

Ramchandani, C. (1974). *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. PhD thesis. MIT, Cambridge, February 1974.

Saïdouni, D. E. (1996). *Sémantique de maximalité : Application au raffinement d'actions en LOTOS*. PhD thesis. LAAS-CNRS, Toulouse, France.

Saïdouni, D. E., & Belala, N. (2006). Actions duration in timed models. *Proceedings of International Arab Conference on Information Technology (ACIT'2006)*. December 19-21th 2006.

Saïdouni, D. E., Belala, N., & Bouneb, M. (2008). Aggregation of transitions in marking graph generation based on maximality semantics for petri nets. *In proceding of the Second International Workchop on Verification and Evaluation of Computer and Communication systems(VeCoS'2008),Leeds,UK. eWiC Series, The British Computer Ssociety (BCS), (ISSN:1477-9358)*. July 2-3th 2008.

Saïdouni, D. E., Belala, N., & Bouneb, M. (2009). Using maximality-based labelled transition system as a model for petri nets. *The International Arab Journal of Information Technology (IAJIT'2009), 5(6):440-446(ISSN: 1683-3198.)*. November 2009.

Saïdouni, D. E., & Courtiat, J. P. (2003). Prise en compte des durées d'action dans les algèbres de processus par l'utilisation de la sémantique de maximalité. *In Proceedings of CFIP2003, France, Hermes*.

Sifakis, J. (1977). Use of petri nets for performance evaluation. *Modelling and Performance Evaluation of Computer Systems*. pp 75–93.

Walter, B. (1983). Timed petri nets for modelling and analyzing protocols with real-time characteristics. *Protocol Specification, Testing and Verification III*. pp 149–159.

Yovine, S. (1997). KRONOS: a verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer*. 1,123–134.

[i] We suppose that $\tau \notin Act$ (τ indicates invisible action, also known as silent or internal action).

[ii] We suppose that all nets are finite, i.e. $|P \cup T| \in \mathbb{N}$.

[iii] D can be $\mathbb{Q}^+$ or $\mathbb{R}^+$.

[iv] The reader may see that when bound tokens become free all their related clocks may not be reused until the associated tokens are consumed.

[v] In practice, we will often use an ordered set, for example $E \subset \mathbb{N}$, provided with the relation $\leq$, the function *get* gives the smallest element of this set.

[vi] The resulting DATA has 36 states and 73 transitions. For this reason, we represent only a fragment of this structure.

[vii] Note that the strong semantics, which represents T-TPN and P-TPN (Merlin, 1974; Khansa, 1997), forces a transition to be fired when it reaches the upper bound of its firing interval. Whereas the lazy semantics, which represents A-TPN and TdPN (T-TdPN, P-TdPN) (Walter, 1983; Ramchandani, 1974; Sifakis, 1977), never forces a transition to be fired, i.e., a token can remain infinitely in a place.