We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

9

A new Control Synthesis Approach of P-Time Petri Nets

Bonhomme Patrice
University François Rabelais of Tours, Laboratoire d'Informatique, 64 avenue Jean Portalis, 37200 TOURS France

1. Introduction

Petri nets (PN) (David & Alla, 1994 ; Murata, 1989) are recognized as an appropriate tool for the modeling and analysis of asynchronous, discrete event systems with concurrency. Originally proposed as a causal model explicitly neglecting time, they have been extended and adapted in several ways in order to fulfill the requirements of specific application areas (Vand der Aalst, 1993 ; Diaz & Senac, 1994 ; Khansa*, et al.* 1996 ; Merlin & Faber, 1976 ; Roux & Déplanche, 2002).

In particular for systems whose functionalities are defined with respect to time (Berthomieu & Diaz, 1991 ; Bonhomme, 2006 ; Bonhomme *et al*., 2001 ; Bucci & Vivario, E., 1995 ; Calvez *et al*., 2004 ; Jiroveanu *et al*., 2006 ; Wang *et al*., 2000) and whose correctness can only be proved by taking time into consideration, PNs are extended with two time parameters representing the minimum and the maximum delay. For Merlin's model (Merlin & Faber, 1976), also called T-time Petri nets or time Petri nets (TPN on short), this interval of firability is associated with each transition of the model representing its static earliest and latest firing time. For P-time Petri nets (P-TPN) (Khansa*, et al.* 1996), each place is associated with a time interval representing the operation duration of a token in this place. To ensure the safe behavior of such systems there is a need for formal approach to model and analyze their correctness.

The idea behind the control concept is to ensure that the system satisfies a set of imposed specifications, its behavior is then restricted to an acceptable one. From a supervisory control point of view, it means that the system never enters a forbidden state or, being given a set of legal words (a set of firing sequence), no illegal word is generated. Inspired by the well-known event-feedback control of Ramadge and Wonham (Ramadge & Wonham, 1989), recently, (Guia *et al*., 2004) proposed a state-feedback control with event observer and initial macromarking (the distribution of the tokens is partially known). The safety specifications under consideration is the limitation of the weighted sum of markings in subsets of places of the studied PN, called generalized mutual exclusion constraints (GMEC). In their approach, they focus on T-timed Petri nets and developped a control scheme with a deadlock recovery procedure by means of observers.

Concerning time models many control techniques are based on linear algebra and thus, are restricted to subclasses of PNs (such as marked graphs, state machines, ...). Similarly, (Freedman, 1991) proposed the structural analysis of a subclasss of T-time Petri nets called Ω which are either deterministic or exhibiting a particular type of conflict called choice among alternatives. (Sathaye & Krogh, 1998) proposed an extension of T-time Petri nets called controlled time Petri nets (CtlTPNs) to model the dynamics of real-time discrete event systems. To fully represent the logical behavior of CtlTPNs control class graphs (CCGs) are also defined. Thanks to this graph (which is an extension of the state classes graph of (Berthomieu & Diaz, 1991) dealing with the control effects) a real-time supervisor, based on a nondeterministic logical supervisor for the CCG, is designed ensuring that the desired specifications are satisfied. However, the method is restricted to bounded CtlTPNs for which a finite CCG exists. More recently, an original approach (Wang *et al.*, 2007) based on PN unfoldings was proposed to enforce transitions deadline in TPN but only safe models are considered.

This paper introduces a new control approach based on the analysis of the P-time Petri Net model structure and more specifically on the set of feasible firing sequences of the underlying untimed Petri net. Furthermore, the proposed approach is not restricted to sub-classes or safe time Petri nets. So, thanks to the determination of an inequalities system generated for a possible evolution of the autonomous model considered, the performances evaluation and the determination of an associated control for a definite functioning mode for the time model are made possible. Thanks to the introduction of partial order on the execution of particular events the developed approach of control is more reactive. Indeed, it is proposed to not impose precedence constraints among operations which can occur concurrently, leading to a flexibility gain (on the operations) in the determination of a feasible control as the scheduling of particular events can be modified. It can be noticed that the idea of not considering the firing order into the timing constraints, for concurrent events, was proposed by (Lilius, 1999). Thanks to a new semantics for time PN, the application of partial-order theory, originally developed for untimed Petri nets, is made possible for TPN. However, its method is restricted to contact-free TPN.

Moreover, the inequalities system is written just once, (it can be done for symbolic values – the inequalities are then written in terms of all the parameters, symbols other than numerics) so, changing the timing constraints do not modify its form and its non-emptiness can also be used to answer questions about reachability of particular markings.

The paper is organized as follows: an informal discussion of the introduction of time in Petri nets is realized in the next section. A formal definition of P-time Petri nets is given in the third one. Section four recalled the material required for the approach, originally devoted to an exhaustive simulation purpose. The proposed approach is then presented in section five and an illustrative example consisting in the supervision of a control distributed in several Programmable Logic Controllers is proposed in the sixth one.

Finally, in the last section some conclusions and future work are presented

2. Petri nets and time

It is assumed, in the following, that the reader is familiar with Petri nets. If it is not the case, please refer to (Murata, 1989) for the basic definitions and terms. There exists several

extensions of Petri nets dealing with time in the literature, each one being dependent on the application considered and are aimed at expressing different types of constraints.
The introduction of timing issues in Petri nets drives to several extended models. The first extension of PN with time was introduced by Ramchandani (Ramchandani, 1973). In his model, called T-timed PN, a delay is associated with each transition representing the duration of an operation. They have been extensively used in a performances evaluation context (Zuberek, 2000). Another timed model proposed by Sifakis (Sifakis, 1977), called p-timed PN associates to each place of the net a delay representing the sojourn time of a token in this place. It was demonstrated that both models are expressively equivalent.
In the timed models a token can wait indefinitely in a place. Indeed, a transition is not forced to fire unless the decision to fire it is explicitely made. Consequently, they are not suitable for the modelling of time critical systems, for which a minimum and a maximum sojourn time are imposed.
Among the time models, the Merlin's model is the most commonly used but others time models, dealing with operation durations specified as intervals, can be found in the literature.
Van der Aalst (Van der Aalst, 1993) proposed the interval timed coloured Petri net (ITCPN), its main feature is the time mechanism associated with each token of the net. Indeed, in this model a timestamp is attached to every token. This timestamp indicates the instant at which a token becomes available. The enabling time of a transition is the maximum timestamp of the tokens to be consumed. Transitions fire as soon as possible and the transition with the smallest enabling time will fire first. Firing is an atomic action which produced tokens with a timestamp of at least the firing time. The difference between the firing time and the timestamp of such a produced token is called the firing delay and it is specified by an interval.

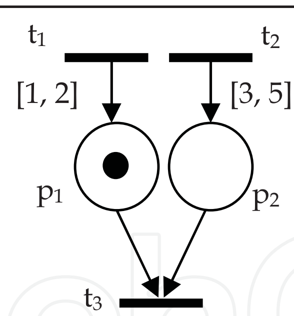

Fig. 1. Portion of ITCPN

For instance, consider the portion of ITCPN above, if transition t_1 is fired at time 0, its firing creates the token in place p_1 with a timestamp within the interval [1, 2]. So, the token contained in place p_1, which input the synchronization transition t_3, can wait indefinitely because this transition will be forced to fire only when the token created by the firing of transition t_2 will become available (at the time: (firing instant of t_2) + x, with $x \in [3, 5]$). To avoid the previous situation, a transformation can be realized and it drives to the representation shown in Fig. 2.

Furthermore, it must be specified that when a conflicting situation occurs the priority is given to the firing of transition t_3, but this particular policy contradicts the fact that the transition with the smallest enabling time should fire first.

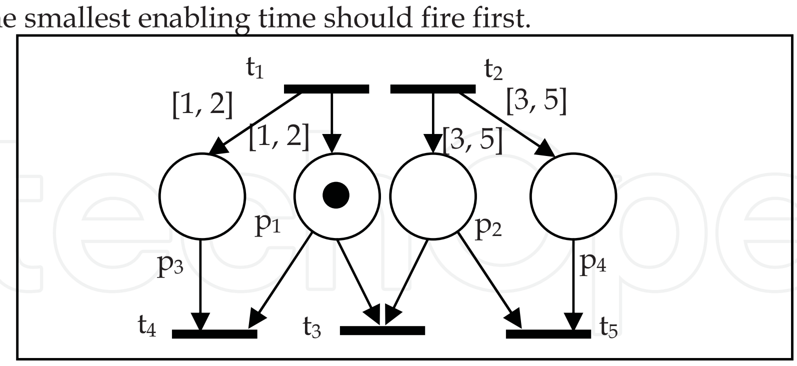

Fig. 2. Transformation of ITCPN.

Time stream Petri nets (TSPN) (Diaz & Senac, 1994) was presented as an extension of the Merlin's model aiming at specifying synchronization constraints in distributed asynchronous multimedia systems and applications. Informally speaking, in this model a time validity interval $[\alpha_i, \beta_i]$ is associated to each arc linking a place p to a transition t. This interval means that if a token arrives in place p at time τ, transition t is force to fire in the interval $[\tau + \alpha_i, \tau + \beta_i]$.

The main feature of this model is the representation of several synchronization mechanisms. Indeed, a complete set of firing rules is proposed to accurately enforce actual synchronization policies between different and related multimedia streams. The "pure-and" rule allows to represent satisfactorily the synchronization mechanism but the different semantics associated with the multiple enabledness of transitions drive to a firing rule which can be sometimes hard to handle.

Consider the following portion of TSPN where the tokens in place p_1 and p_2 are arrived at time 0. The enabledness intervals of transition t_1 associated with the three synchronization modes, strong-or, weak-and and pure-and, are also represented.

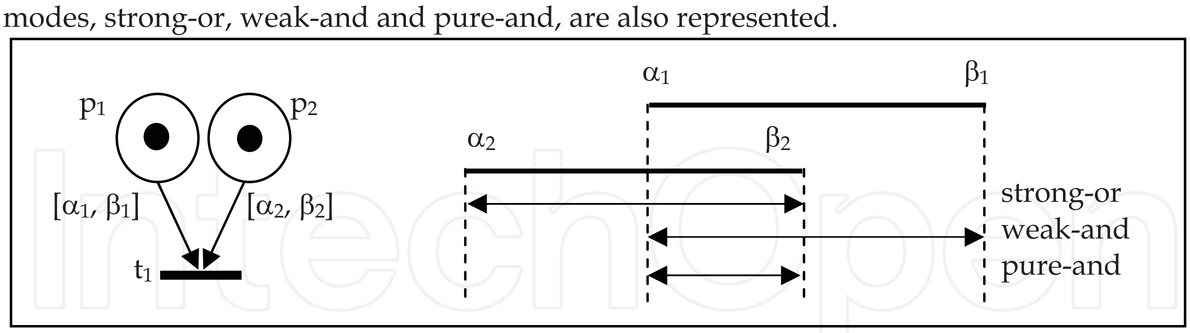

Fig. 3. Time stream Petri net and synchronization.

(Ghezzi *et al.*, 1991) proposed a high-level model called Entity-Relation net (ERN), as the one developed by Van der Aalst, each token carries a set of informations and each transition is associated with relations. These relations allow to select the different tokens involved in the enabledness of particular transitions and they also indicate the type of the tokens which can be created (depending on the consumed ones).

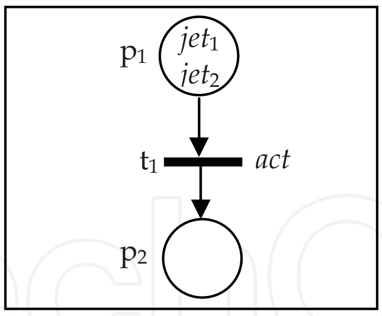

Fig. 4. Portion of ER net.

For instance, in Fig. 4, place p_1 contains two tokens jet_1 = {<x, 1>, <y, 1>} and jet_2 = {<x, 0>, <y, 1>}, the relation attached to transition t_1 is act = {<p_1, p_2> | $p_1.x = p_1.y$ and $p_2.x = p_1.x$ and $p_2.y \in \{z \mid p_1.y \le z \le p_1.y + p_1.x\}$}. In the previous relation p_1 means any token in this place (in this case, either jet_1 or jet_2) and <x, 1> means $p_1.x = 1$. The firing of transition t_1 is represented by the expression <p_1, t_1, p_2> or equivalently when it is clear from the context <p_1, p_2>. The token jet_1 is satisfying act while jet_2 is not, so, the token created by the firing of t_1 denoted as p_2 will verify $p_2.x = 1$ and $p_2.y \in [1, 2]$ and the token jet_2 will be considered as dead (thus, it can be removed from the net).

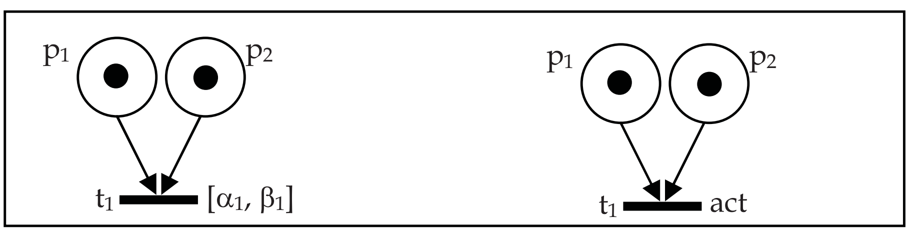

Fig. 5. Merlin's model / Time ER model.

In this model, the time factor can be introduced thanks to a timestamp (denoted as *chronos*) associated to each token of the net and representing its creation instant – the time ER model (TER) is then obtained. Originally, a weak time semantic is associated with this model (i.e. transitions are not force to fire) but the authors proposed a non trivial transformation rule allowing to consider a strong time semantic.

For instance, consider the two portions of PN (Merlin's model / time ER model) depicted on Fig. 5, both synchronization transitions will have the same behaviour if the transition t_1 of the TER model is associated with the following relation act:

$$act = \{x \mid \max\{p_1.chronos, p_2.chronos\} + \alpha_1 \le x \le \max\{p_1.chronos, p_2.chronos\} + \beta_1\}.$$

In the next section, the chosen modeling tool, the P-time Petri nets is presented

3. P-Time Petri nets

The formal definition of a P-TPN (Khansa *et al.*, 1996) is given by a pair < Nr; I > where:

Nr is a marked Petri Net (David & Alla, 1994)

$I: P \rightarrow (Q^+ \cup \{0\}) \times (Q^+ \cup \{\infty\})$

$p_i \rightarrow I_i = [a_i, b_i]$ with $0 \le a_i \le b_i$

Where:

$\boldsymbol{P}$: the set of places of net *Nr,*

$\boldsymbol{Q}^+$: the set of positive rational numbers.

I_i defines the static interval of the operation duration of a token in place p_i.

A token in place p_i will be considered in the enabledness of the output transitions of this place if it has stayed for a_i time units at least and b_i at the most. Consequently, the token must leave p_i, at the latest, when its operation duration becomes b_i. After this duration b_i, the token will be "dead" and will no longer be considered in the enabledness of the transitions. Notice that: a dead token is not removed from the place, this token state indicates that a potential time violation has occurred. As the death of tokens generally occurs in places which are input places of a synchronization transition an algebraic approach using (min, max, +) algebra was proposed to check their correct behaviour for P-time marked graphs (Declerck & Alaoui, 2004).

The particularity of this model requires analysis techniques, allowing taking account efficiently of the various functionalities associated with the modeled system, as well as its time features. It leads ineluctably to the need for having formal methods ensuring the system control. Indeed, the policy consisting in firing a transition as soon as it becomes enabled is not always feasible and usually leads to a potential constraint violation.

It can be noticed that for P-TPN the contribution of each token present in the net must be taken into account because it must be prevented from dying whether it participates to the enabledness of a transition or not. In the Merlin's model the situation is not the same. Indeed, consider the following synchronization mechanism for the Merlin's model depicted on Fig. 6.

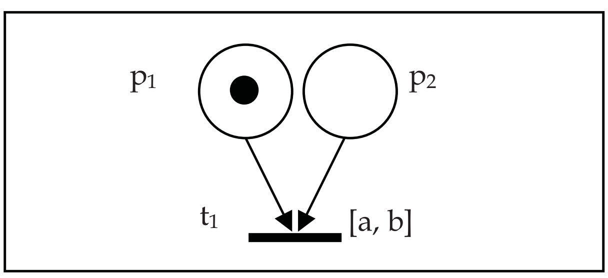

Fig. 6. Synchronization and T-time PN

The token in place p_1 can wait indefinitely and the local clock associated with transition t_1 is triggered when a token arrives in place p_2.

For a net *Nr* the conventional dot notation which can naturally be extended to set of nodes, will be used:

- °t (t°) = the set of input (output) places of transition t.
- °p (p°) = the set of input (output) transitions of place p.

The following section is devoted to the control approach, based on the evaluation of the firing condition.

4. Firing instant notion

Definition 1 (Boucheneb & Berthelot, 1993)*:* A fired transition denoted by t_j will be associated with the j^{th} firing instant (i.e. the firing sequence considered is $t_1t_2t_3..t_q$). A variable x_i will represent the elapsed time between the $(i - 1)^{th}$ and the i^{th} firing instant.

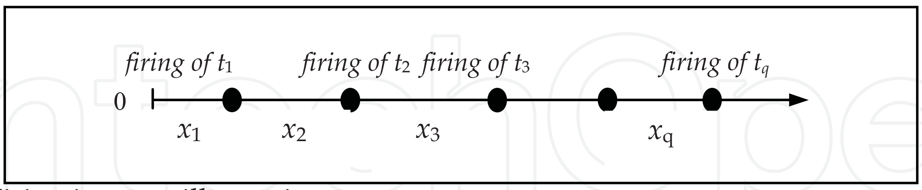

Fig. 7. Firing instants illustration

For instance on Fig. 7, $(x_2 + x_3)$ is the time elapsed between the first firing instant and the third one.

In a P-TPN, the sojourn time (i.e. the amount of time that a token has been waiting in a place) is counted up as soon as the token has been dropped in the place as seen previously. Thus, quantitative (i.e. performance) considerations take precedence over qualitative (i.e. logical) ones, in opposition to Merlin's time PN model. To compute the firing instants, this approach requires that a token is identified by three parameters: the place that contains it, the information of its creation instant and the information of its consumption one.
Function *TOK* is defined with this purpose. When the weight of the P-TPN arcs is element of **N**, *TOK*(*j*, *n*) is a multi-set.
For the sake of simplicity, only P-TPN with arcs weight element of {0, 1} are considered here. Thus:

$$TOK: \mathbf{N} \times \mathbf{N}^* \to \wp(P) \quad \text{(with } \mathbf{N}^* \text{ the set of strictly positive natural numbers)},$$

$TOK(j, n) = \{p \in P \ / \ p$ contains a token created by the j^{th} firing instant and consumed by the n^{th} one in $\sigma\}$.
With:

$\wp(P)$ the set of subsets of ***P***,

and σ a considered firing sequence.

Several tokens contained in the same place will be differentiated by the values *j* and *n* associated with them. So, it is possible to impose any token management, but in the sequel a FIFO mode will be considered for the sake of simplicity. Moreover, the determination of these sets is closely linked to the firing sequence considered.

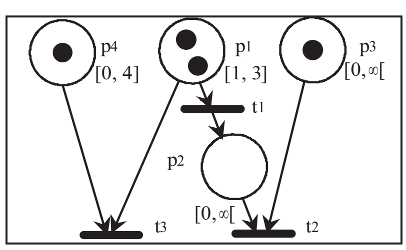

Fig. 8. Illustration of set *TOK*

For instance, the firing sequence considered is $t_1t_2t_3$. The following sets are obtained:

$TOK(0, 1) = \{p_1\}$, $TOK(0, 2) = \{p_3\}$,
$TOK(1, 2) = \{p_2\}$, $TOK(0, 3) = \{p_1, p_4\}$.

Using these sets, the minimal and maximal effective sojourn times of each token in its place are evaluated by:

$$Dsmin(j, n) = \begin{cases} \max(a_i),\ p_i \in TOK(j,n), \\ \text{else } 0 \text{ if } TOK(j,n) = \varnothing \end{cases}$$

$$Dsmax(j, n) = \begin{cases} \min(b_i),\ p_i \in TOK(j,n) \\ \text{else } +\infty \text{ if } TOK(j,n) = \varnothing \end{cases}.$$

Indeed, tokens, with the same creation instant, located in different places and involved in the same transition firing may mutually constrained their (static) sojourn time.
For instance, consider the portion of P-TPN depicted in Fig. 8, $Dsmin(0,3) = \max(0,1) = 1$ and $Dsmax(0,3) = \min(3,4) = 3$ and the initial token in place p_4 is affected by the token contained in place p_1 as they are involved in the firing of transition t_3.
As we are intrested in cyclic behavior, the next paragraph deals with the periodic functioning mode.

4.1 Time periodic control

The behavior of this mode is fully determined by:

$$\forall k \geq 1,\ s_i(k) = s_i(1) + (k - 1).\pi \qquad (1)$$

Where $s_i(k)$ is the k^{th} firing date of the transition t_i and π the cycle time (or the functioning period). That means that the times of the first firing of the transitions and the functioning period are sufficient to entirely describe the periodic functioning mode. So, a periodic time schedule can be built.
The behavior of a P-TPN can be discribed in terms of a firing schedule.

Definition 2: A P-TPN *Nr* firing schedule, will be a sequence of ordered pairs $(t_i, \sum_{k=0}^{i} x_k)$; transition t_i firable at time $(\sum_{k=0}^{i} x_k)$, obtained from the state reached by starting from *Nr* initial state and firing the transitions t_j, $1 \leq j < i$, in the schedule at the given times. It can be noticed that in order to facilitate the understanding, it is considered that the studied firing sequence is the following: $\sigma = t_1 t_2 t_3 \ldots t_q$, thus the r^{th} event is related to the firing of transition t_r at the absolute time $\sum_{k=0}^{r} x_k$.

The next section deals with the control approach.

5. Control approach

5.1 Interval arithmetic

Let $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$, with $0 \le a_1 \le b_1 \le +\infty$ and $0 \le a_2 \le b_2 \le +\infty$. Then $I_1 + I_2$ can be defined as the interval $[a_1 + a_2, b_1 + b_2]$ and $I_1 - I_2$ as the interval $[a_1 - a_2, b_1 - b_2]$ if $a_2 \le a_1$ and $b_2 = a_2$. Let IV be the interval $[\alpha, \beta]$, IV^{min} will represent α and IV^{max} will represent β.
In this subsection, for the sake of simplicity, it will be considered that, starting from the initial state the firing of an enabled transition t_i will lead to the state labelled by i (the i^{th} state).
The principle of the proposed control approach will be first illustrated on the following simple example.
At the initial state, the interval of availability of each token in place p_i, $DS_0(p_i)$ can be evaluated as follows:

$$\forall p_i \in TOK(0, .), DS_0(p_i) = [a_i, b_i],$$
$$\text{and } MIN_0 = Dsmax(0, 1),$$

On the P-TPN model of Fig. 9:

$$DS_0(p_1) = [4, 6], DS_0(p_3) = [3, 5] \text{ and } MIN_0 = 5.$$

DS_0 contains the static intervals of all the tokens initially present in the net and MIN_0 represents the time after which a time constraint is ineluctably violated leading to the death of a token.

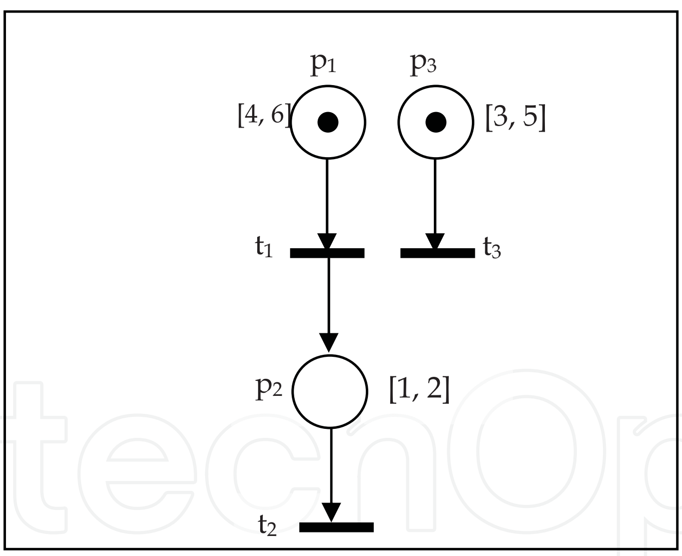

Fig. 9. Illustration.

An enabled transition t_1 is said to be fireable if:

t_1 is enabled in the autonomous PN sense (i.e. $M_0 \ge \text{Pre}(., t_1)$, with M_0 the initial marking)
and
$Dsmin(0, 1) \le MIN_0$.

On the P-TPN model of Fig. 9, transition t_1 is fireable because there is one token in place p_1 and $DS_0^{\min}(p_1) \leq 5$.

After the firing of transition t_1, two situations should be considered:

1. for the newly created token(s) (created by the first firing instant, the firing of transition t_1):

$$\forall p_i \in TOK(1, .), DS_1(p_i) = x_1 + [a_i, b_i],$$

2. for the token(s) created initially and not involved in the firing of t_1:

$$\forall p_i \in TOK(0, j), j \neq 1, DS_1(p_i) = [\max(x_1, a_i), b_i],$$

$$MIN_1 = \min(DS_1^{\max}(p_i)), \forall p_i \in TOK(0, j) \cup TOK(1, k), j \neq 1 \text{ and } k > 1.$$

On the P-TPN model of Fig. 9:
The token created in place p_2 by the firing of t_1 (at time x_1) will be considered in the enabledness of transition t_2 within the interval $DS_1(p_2) = x_1 + [1, 2] = [x_1 + 1, x_1 + 2]$.
For the token in place p_3, created initially and not involved in the firing of t_1, the minimal time at which it could participate to the firing of transition t_3 must be updated.
For instance, if transition t_1 is fired at time $x_1 = 4$, as the firing of t_3 will occur after the firing of t_1, the token in place p_3 will be considered at the earliest (in the firing of transition t_3) at time $max(x_1, 3) = 4$, so $DS_1(p_3) = [4, 5]$.
A generalization of the previous principle is realized in the following:

For a given firing sequence $\sigma = t_1 t_2 t_3 \ldots t_q$ (its length corresponding to $|\sigma| = q$) (M_i the marking reached by the firing of t_i) the lower (resp. upper) bound denoted $\mu_\sigma^{\min}$ (resp. $\mu_\sigma^{\max}$) of its average cycle time can be computed by the intermediary of the following linear programs, stated as follows:

$$\mu_\sigma^{\min} = \text{minimize}(\pi) \text{ and } \mu_\sigma^{\max} = \text{maximize}(\pi),$$

$$\text{with } \pi = \sum_{i=0}^{|\sigma|} x_i ,$$

subject to the set of constraints:

$$\sum_{i=0}^{j} x_i \in INTV_j, \forall j \in \{0, \ldots, |\sigma|\}.$$

With $INTV_j, \forall j$ obtained as follows:

$$INTV_1 = [Dsmin(0, 1), MIN_0],$$

Evaluation of the set DS_1:

- contribution of token(s) created by the firing of t_1 (the first fired transition):

$$\forall p_i \in TOK(1, .), DS_1(p_i) = x_1 + [a_i, b_i],$$

- contribution of token(s) created initially and not involved in the firing of t_1:

$$\forall p_i \in TOK(0, j),\ j > 1,\ DS_1(p_i) = [\max(x_1, a_i), b_i],$$
$$\text{and } MIN_1 = \min(DS_1^{\max}(p_i)), \forall p_i \in M_1.$$

The case $i > 1$:

From the marking M_{i-1} the firing of transition t_i leads to the following sets:

$$INTV_i = [\max(\boldsymbol{DS}_{\boldsymbol{i-1}}^{\min}(\boldsymbol{p_k})), MIN_{i-1}],$$
$$\forall p_k \in TOK(., i),$$

Evaluation of the set DS_i:

1. the newly created token(s):

$$\forall p_j \in TOK(i, .),$$
$$DS_i(p_j) = \sum_{u=0}^{i} x_u + [a_j, b_j],$$

the token(s) created by the s^{th} firing instant, with $s < i$, which are not involved in the firing of t_i:

$$\forall p_k \in TOK(s, j),\ j > i,$$
$$DS_i(p_k) = [\max(\sum_{u=0}^{i} x_u , \sum_{u=0}^{s} x_u + a_k), \sum_{u=0}^{s} x_u + b_k],$$

and $MIN_i = \min(DS_i^{\max}(p_k)), \forall p_k \in TOK(., j),\ j > i.$

5.2 Concurrency considerations

It can be noticed that a strict ordering of the events is imposed: if the obtained inequalities system is written for a firing sequence $\sigma = t_1t_2t_3\dots t_q$, its emptiness means that the considered sequence is not feasible.

Consider the P-TPN of fig. 10 and the firing sequence $\sigma = t_1t_2$, the system S_σ is:

$$\boldsymbol{S_\sigma} \begin{cases} 4 \le \boldsymbol{x}_1 \le \min(6, 3) \\ \max(\boldsymbol{x}_1, 1) \le \boldsymbol{x}_1 + \boldsymbol{x}_2 \le 3 \end{cases}$$

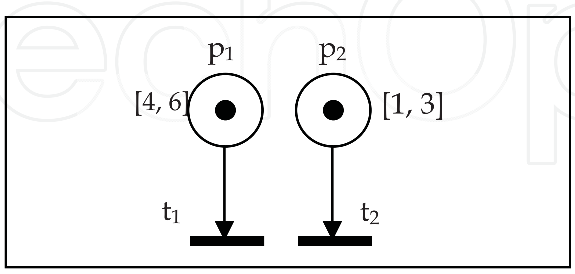

Fig.10. P-TPN with concurrency.

On this simple illustration, due to the timing constraints it is obvious that this system admits no solution traducing the fact that transition t_1 cannot be fired before transition t_2.

In the system obtained previously a relation of precedence was imposed on the firing instants of the two concurrent transitions $t_1$and t_2. This relation was explicitly exhibited by the inequality $\max(x_1, 1) \leq x_1 + x_2$ i.e. the firing of t_1 must occur before the firing of t_2.
It is proposed to not imposed precedence constraints on the firing of parallels events and no longer consider the constraints induced by token(s) not involved in the firing instant considered which initiate another firing.
For the example of Fig. 10, it yields to the new following system:

$$\boldsymbol{S_\sigma} \begin{cases} 4 \leq \boldsymbol{x}_1 \leq 6 \\ 1 \leq \boldsymbol{x}_1 + \boldsymbol{x}_2 \leq 3 \end{cases}$$

for instance if the firing instant of transition t_1 is $x_1 = 6$, it follows: $-5 \leq x_2 \leq -3$, this inequality means that the firing of transition t_2 must occur before the firing of t_1 and its firing instant corresponds to x_1+x_2. Its earliest one is 6 + (-5) = 1 and its latest one is 6 + (-3) = 3.
So, an example of schedule is:

$$\varpi = ((t_2, x_1 + x_2 = 6\text{-}5 = 1), (t_1, x_1 = 6)).$$

It can be noticed that the ordering of the events in the schedule considered will be made on the basis of the value of the sum $\sum_{k=0}^{i} x_k$.
More formally, the set *TS*, defined as follows need to be introduced:

TS: $\boldsymbol{N} \rightarrow$ subset of (*T*),
$i \rightarrow \{t \in T \mid t$ is enabled by the i^{th} firing instant and t is a persistent transition$\}$.
With *T* the set of transitions of the net and $\boldsymbol{N}$ the set of naturals numbers. An enabled transition is called a persistent transition if it can be disabled only by its own firing.
For the evaluation of each firing instant the following procedure must be applied.
Consider the current fired transition of the considered sequence to be t_i. So, the net is in the state reached after the firing of the i^{th} fired transition.
Let $TS(i) = \{t_{k1}, t_{k2}, t_{k3}, ..., t_{kn}\}$ ($|TS(i)| = n$, i.e. the cardinal of the set).

The firing of transition t_{kj} with $j = 1, ..., n$ will be associated with the $(i + j)^{\text{th}}$ firing instant.

$$\forall u = i + 1, ..., i + n, \forall p_v \in TOK(., u),$$

$$INTV_u = [\max(\boldsymbol{DS}_{\boldsymbol{u}-1}^{\min}(\boldsymbol{p_k})), \boldsymbol{MIN}_{\boldsymbol{u}-1}\big|\boldsymbol{p_r} \in \boldsymbol{TOK}(., \boldsymbol{u}) \cup \boldsymbol{TOK}(., \boldsymbol{s})],$$

$$\text{with } s > i + n.$$

$\boldsymbol{MIN}_{\boldsymbol{s}}\big|\boldsymbol{p_r} \in \boldsymbol{P}'$ with $\boldsymbol{P}' \subset \boldsymbol{P}$ is the restriction of *MINs* to the set of places *P'* (i.e. *MINs* is evaluated by considering only the contribution of the places of the set *P'*).
The preceding expression allows to consider, for a particular firing instant, the tokens which participate to this firing and those present at the evaluation instant but which are not involved in the enabledness of a transition (they must be prevented from dying). Thus, the tokens involved in the firing of a parallel transition are not considered.
The benefits of this approach are patent because in presence of concurrency, it is not necessary to take into consideration the set of all possible interleavings. An "untimed firing sequence" (feasible on the untimed underlying Petri net model) will be considered and via

linear programming techniques the solution obtained by solving the obtained system (associated with the firing sequence considered) will determine the "real" order of the events and the exact duration of the resulting firing sequence. For instance, consider the case of n transitions which can concurrently fire, from a state. The application of the presented method will result in an unique inequations system instead of $n!$ systems, each one being associated with a possible combination of the order of the events.
Furthermore, this approach can be used to bring answers to solve reachability problems. Indeed, by considering the possible evolutions from a source marking and leading to a target marking, in terms of firing sequence, a set of inequalities systems can be obtained. So, if this set of systems admits no solution it means that there is no feasible firing sequence on the P-TPN model allowing to, starting from the source marking, reach the target one. It is well known that enumerative based methods usually face the so-called state space explosion problem. Although the investigation of the feasible sequences is realized on the untimed model, the approach may be hampered by the complexity inherent to this kind of procedure.

6. Illustrative example

In this section, an example is used to illustrate the developed approach. The background is the supervision of a control distributed in several Programmable Logic Controllers (PLC on short).
The supervision consists of accessing via a network to control system variables (temperature, pressure, level in a tank,...) by means of different requests to the PLCs. The duration of requests depends on many factors as medium and protocol used, PLC activity, number of variables to be read or write, ... Generally, these durations are known not exactly but by a time interval.
However, ensuring the consistency of the controlled variables to be spied and/or modified requires that the duration between different requests does not exceed a defined duration imposed by the dynamic of the observed process.
Then, the consistency constraint is not a time-out one, but an operation time one (its latency can be stated by means of a time interval in a general case).
Consequently, to model naturally the repetitive functioning of the supervisor, a P-time PN model can be built. Fig. 11 represents a scheduling problem of a supervisor.
This supervision consists of making 6 different requests (p_{12}, p_{14}, p_{22}, p_{25}, p_{32}, p_{34}) to the three PLCs (p_{10}, p_{20}, p_{30}).
This model takes into account:

- Precedence constraints (e.g. by means of (p_{12}, t_{12}, p_{13}, t_{13}, p_{14}) the request associated with p_{14} must be processed after the one associated with p_{12}).
- Synchronization constraints, e.g. by means of t_{40}, all the requests must be processed to ensure the consistency constraint (no request can be processed twice without processed all the others).
- Shared resources constraints: the PLC associated with p_{20} cannot process the requests p_{22} and p_{25} at the same time.

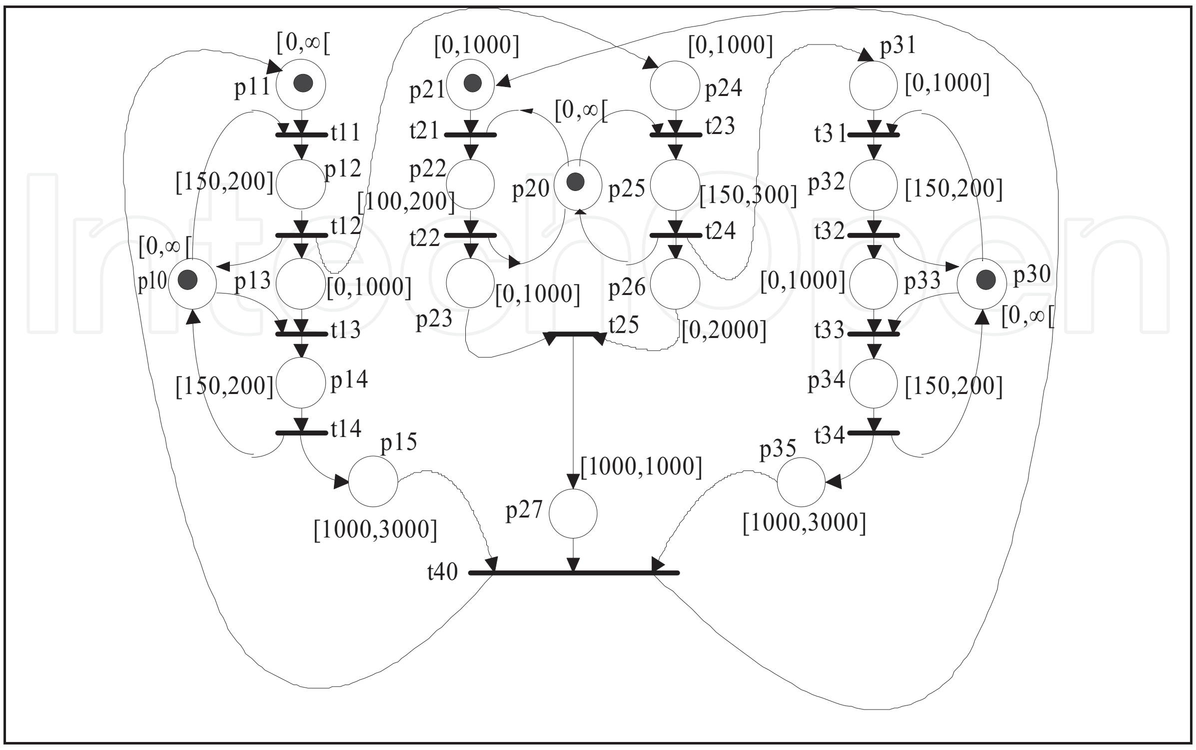

Fig. 11. P-TPN model

t_{11}	start processing request 1
t_{12}	completion of request 1
t_{13}	start processing request 2
t_{14}	completion of request 2
t_{21}	start processing request 3
t_{22}	completion of request 3
t_{23}	start processing request 4
t_{24}	completion of request 4
t_{25}	consistency of request 3 with request 4
t_{31}	start processing request 5
t_{32}	completion of request 5
t_{33}	start processing request 6
t_{34}	completion of request 6
t_{40}	global consistency

Table 1. Meaning of transitions of Fig. 11

Consider the firing sequence $\sigma = t_{11}t_{12}\ t_{13}t_{14}\ t_{21}t_{22}\ t_{23}t_{24}\ t_{25}t_{31}\ t_{32}t_{33}\ t_{34}t_{40}$, the simplified following inequalities system is then obtained:

$$0 \le x_1 \le \infty$$
$$150 \le x_2 \le 200$$
$$0 \le x_3 \le 1000$$
$$150 \le x_4 \le 200$$
$$0 \le \sum_{i=1}^{5} x_i \le 1000$$
$$\sum_{i=1}^{5} x_i + 100 \le \sum_{i=1}^{6} x_i \le \sum_{i=1}^{5} x_i + 200$$
$$\max(\sum_{i=1}^{6} x_i, \sum_{i=1}^{2} x_i) \le \sum_{i=1}^{7} x_i \le \min(\sum_{i=1}^{5} x_i + 200, \sum_{i=1}^{4} x_i + 3000)$$
$$\sum_{i=1}^{7} x_i + 150 \le \sum_{i=1}^{8} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{7} x_i + 300)$$
$$\max(\sum_{i=1}^{6} x_i, \sum_{i=1}^{8} x_i) \le \sum_{i=1}^{9} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{6} x_i + 1000, \sum_{i=1}^{8} x_i + 2000)$$
$$\sum_{i=1}^{8} x_i \le \sum_{i=1}^{10} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{8} x_i + 1000, \sum_{i=1}^{9} x_i + 1000)$$
$$\sum_{i=1}^{10} x_i + 150 \le \sum_{i=1}^{11} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{9} x_i + 1000, \sum_{i=1}^{10} x_i + 200)$$
$$\sum_{i=1}^{11} x_i \le \sum_{i=1}^{12} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{9} x_i + 1000, \sum_{i=1}^{11} x_i + 1000)$$
$$\sum_{i=1}^{12} x_i + 150 \le \sum_{i=1}^{13} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{9} x_i + 1000, \sum_{i=1}^{12} x_i + 200)$$
$$\max(\sum_{i=1}^{4} x_i + 1000, \sum_{i=1}^{9} x_i + 1000, \sum_{i=1}^{13} x_i + 1000) \le \sum_{i=1}^{14} x_i$$
$$\sum_{i=1}^{14} x_i \le \min(\sum_{i=1}^{4} x_i + 3000, \sum_{i=1}^{9} x_i + 1000, \sum_{i=1}^{13} x_i + 3000)$$

Let $\pi = \sum_{i=0}^{14} x_i$ and $\mu_\sigma^{\min}$ = minimize(π) and $\mu_\sigma^{\max}$ = maximize(π),

subject to the constraints of the previous system lead to $\mu_\sigma^{\min}$ = 1600 and $\mu_\sigma^{\max}$ = 3200 units of time.

For instance, the firing instants (the x_i's) corresponding to $\mu_\sigma^{\min}$ are:

[0, 150, 300, 150, -550, 100, 0, 150, 300, -300, 150, 0, 150, 1000] and the associated schedule is the following: ϖ = ((t_{11}, 0), (t_{21}, 50), (t_{12}, 150), (t_{22}, 150), (t_{23}, 150), (t_{24}, 300), (t_{31}, 300), (t_{13}, 450), (t_{32}, 450), (t_{33}, 450), (t_{14}, 600), (t_{25}, 600), (t_{34}, 600), (t_{40}, 1600)).
It can be noticed that, although the inequalities system was obtained on the basis of firing sequence $\sigma = t_{11}t_{12}\ t_{13}t_{14}\ t_{21}t_{22}\ t_{23}t_{24}\ t_{25}t_{31}\ t_{32}t_{33}\ t_{34}t_{40}$, the real feasible sequence is $t_{11}t_{21}\ t_{12}t_{22}\ t_{23}t_{24}\ t_{31}t_{13}\ t_{32}t_{33}\ t_{14}t_{25}\ t_{34}t_{40}$.
In the case of $\mu_\sigma^{\max}$, the corresponding firing instants (the x_i's) are:

[1600, 150, 300, 150, -1200, 200, 550, 150, 300, -300, 150, 0, 150, 1000] and the associated schedule is the following:
ϖ' = ((t_{21}, 1000), (t_{22}, 1200), (t_{11}, 1600), (t_{12}, 1750), (t_{23}, 1750), (t_{24}, 1900), (t_{31}, 1900), (t_{13}, 2050), (t_{32}, 2050), (t_{33}, 2050), (t_{14}, 2200), (t_{25}, 2200), (t_{34}, 2200), (t_{40}, 3200)).

The inequalities system was obtained for a firing sequence which can be viewed as a "pseudo firing sequence".
Indeed, it is used to determine the logical constraints (precedence, mutual exclusion and/or concurrency) but the real order of the events (i.e. the real firing sequence) is obtained thanks to the value and the sign of the firing instants resulting from the resolution of the considered system.

6. Conclusion and future work

Time has become a major issue in the analysis of production systems. Indeed, for time-critical systems where all taks are time-constrained, this parameter does not affect only the system performances but also its correctness.
Thus, an acceptable behavior of such systems depends not only on the order of the events but particularly on the time at which the results are produced. Consequently, correctness and performance issues are closely linked. Due to time intervals specifications, time critical systems require a time control.
In this paper, a new approach to design a time control has been proposed. It uses P-time Petri net as a modeling tool and it uses the firing instants notion that does not require strong structural properties (it is not restricted to subclasses of PN).
The proposed method is based on a token player algorithm and it investigates the set of feasible firing sequences of the underlying untimed Petri net of the considered P-TPN.
The presented technique yields to the obtaining of an inequalities system written once (at the beginning of the analysis) for symbolic values of the timing constraints associated with each place of the considered time model. The ability of not considering only numerical quantities can also be used in order to test real-time systems specifications.
Indeed, as seen in the illustrative example, the (logical) ordering of particular operations can be imposed by the timing constraints, so if an obtained ordering of the events is inappropriate the timing constraints can be modified without altering the system considered and the new scheduling will result in the resolution of the original system.
An issue currently being investigated is the integration of the presented approach in the evaluation of the robustness of the modelled system.

7. References

Van der Aalst, W. M. P. (1993). Interval Timed Coloured Petri Nets and Their Analysis, In: *Application and Theory of Petri Nets,* M. Ajmone Marsan, (Ed.), 453-472, volume 691 of Lecture Notes in Computer Science, Springer-Verlag

Berthomieu, B. & Diaz, M. (1991). Modeling and verification of time dependent systems using time Petri nets. *IEEE Trans. Soft. Eng,* Vol. 16, No. 3, 259-273

Bonhomme, P. (2006). Control and Performances Evaluation of Time Dependent Systems Using an Enumerative Approach, *Proceedings of Conf. on Control Applications, CCA/ISIC,* Germany, Munich

Bonhomme, P., Aygalinc P. & Calvez, S. (2001). Towards the Control of Time-Critical Systems, *Proceedings of Conf. on Control Applications, CCA/ISIC,* Mexico

Boucheneb, H. & Berthelot, G. (1993). Towards a simplified building of Time Petri Nets reachability graph, *Proceedings of PNPM'93,* pp. 46-55

Bucci, G. & Vivario, E. (1995). Compositional validation of time-critical systems using communicating time Petri nets. *IEEE Trans. Soft. Eng,* Vol. 21, No. 12, 969-992

Calvez, S., Aygalinc P. & Bonhomme, P. (2004). Proactive/Reactive Approach for Transient Functioning Mode of Time Critical Systems, *Proceedings of the Third Conference. on Management and Control of Production and Logistics(MCPL'2004),* pp. 65-70, Santiago, Chili

David, R. & Alla, H. (1994). Petri Nets for Modeling of Dynamic Systems - A survey. *Automatica,* Vol. 30, No.2, 175-202

Declerck, P. & Alaoui, M. (2004). Modelling and analysis of p-time event graphs in the (min, max, +) algebra, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'04),* pp. 1807-1812, The Hague, The Netherlands

Diaz, M. & Senac, P. (1994). Time stream Petri nets: a model for timed multimedia information, *Proceedings of the 15th International Conference on Application and Theory of Petri Nets,* 219-238, volume 815 of Lecture Notes in Computer Science, Springer-Verlag

Freedman, P. (1991). Time, Petri nets, and robotics Robotics and Automation. *IEEE Transactions on Robotics and Automation,* Vol. 7, No. 4, 417 – 433

Ghezzi, C. Mandrioli, D., Morasca, S. & Pezzè, M. (1991). A Unified High-level Petri Net Model for Time Critical Systems, *IEEE Trans. Soft. Eng,* Vol. 17, No. 2, 160 – 172

Giua, A., Seatzu, C. & Basile, F. (2004). Observer-based state-feedback control of timed Petri nets with deadlock recovery. *IEEE Trans. On Automatic Control,* Vol. 49, No. 1, 17-29

Jiroveanu, G., Boel, R. K. & De Schutter, B. (2006). Fault diagnosis for time Petri nets, *Proceedings of the 8th International Workshop on Discrete Event Systems (WODES'06),* pp. 313–318, Ann Arbor, Michigan

Lilius, J. (1999). Efficient state space search for time Petri nets, *Electronic Notes in Theoretical Computer Science,* Vol. 18

Khansa, W., Denat, J. P. & Collart, S. (1996). Fault P-Time Petri Nets for Manufacturing Systems, *Proceedings of Wodes'96,* pp. 19-21, Edinburgh, UK

Merlin, P. & Faber, D. (1976). Recoverability of communication protocols - implications of a theoretical study. *IEEE Trans. Communications,* Vol. 24, No. 9, 381-404

Murata, T. (1989). Petri Nets : Properties, Analysis and Applications. *Proc. of the IEEE,* Vol. 77, No. 4, 541-580

Ramadge, P. J. & Wonham, W. M. (1989). The control of discrete event systems. *Proc. of the IEEE,* Vol. 77, No. 1, 81-98

Ramchandani, C. (1973). Analysis of asynchronous concurrent systems by Petri nets. Ph.D. Thesis, MIT

Roux, O. H.. & Déplanche, A. M. (2002). A t-time Petri net extension for real time-task scheduling modeling. *European Journal of Automation (JESA),* Vol. 36, No. 7, 973-986

Sathaye, A. S. & Krogh. G. H. (1998). Supervisor Synthesis for Real-Time Discrete Event Systems, *Discrete Event Dynamic Systems*, Vol. 8, No. 1, 5-35

Sifakis, J. (1977). Use of Petri nets for performance evaluation, In: *Measuring, Modelling and Evaluating Computer Systems,* H. Beilner and E. Gelenbe, (Ed.), 75-93

Wang, J., Deng, Y. & Xu, G. (2000). Reachability Analysis of real-Time Systems Using Time Petri Nets. *IEEE Tran*s. on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 30, No 5, 725-736

Wang, H., Grigore, L., Buy, U. & Darabi, H. (2007). Enforcing Transition Deadlines in Time Petri Nets, *Proceedings of the 12th IEEE Conf. on Emerging Technologies and Factory Automation, ETFA 2007,* Greece

Zuberek, W. M. (2000) Timed Petri net models of cluster tools, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'2000)*, pp. 3063-3068, Vol. 4, Nashville, TN

Petri Nets Applications

Edited by Pawel Pawlewski

ISBN 978-953-307-047-6

Hard cover, 752 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Petri Nets are graphical and mathematical tool used in many different science domains. Their characteristic features are the intuitive graphical modeling language and advanced formal analysis method. The concurrence of performed actions is the natural phenomenon due to which Petri Nets are perceived as mathematical tool for modeling concurrent systems. The nets whose model was extended with the time model can be applied in modeling real-time systems. Petri Nets were introduced in the doctoral dissertation by K.A. Petri, titled "„Kommunikation mit Automaten" and published in 1962 by University of Bonn. During more than 40 years of development of this theory, many different classes were formed and the scope of applications was extended. Depending on particular needs, the net definition was changed and adjusted to the considered problem. The unusual "flexibility" of this theory makes it possible to introduce all these modifications. Owing to varied currently known net classes, it is relatively easy to find a proper class for the specific application. The present monograph shows the whole spectrum of Petri Nets applications, from classic applications (to which the theory is specially dedicated) like computer science and control systems, through fault diagnosis, manufacturing, power systems, traffic systems, transport and down to Web applications. At the same time, the publication describes the diversity of investigations performed with use of Petri Nets in science centers all over the world.