

Strong Behavioral Similarities in Timed-Arc Petri Nets[☆]

Valentín Valero

School of Informatics, University of Castilla-La Mancha, Albacete, Spain

Abstract

The concept of process similarity has attracted the attention of many researchers in the recent literature, since it measures the degree of proximity of processes. Similarities are therefore useful for the efficient management of large repositories, since they allow us to find the appropriate process models among hundreds or thousands of possible candidates. However, time constraints are usually omitted when these similarity measures are defined, so in this paper two similarity measures are defined over a timed extension of Petri nets, the so-called timed-arc Petri nets, in which tokens are assigned an age indicating the time elapsed from creation, and PT-arcs (place to transition arcs) are labeled with time intervals that are used to restrict the age of the tokens that can be used to fire the adjacent transition.

Keywords: formal methods, timed-arc Petri nets, reachability graphs, similarities.

1. Introduction

A similarity measure provides us with a quantitative information about the degree of proximity in the behavior of two processes. This has been a focus of recent research since process models are extensively used in many areas of current technologies, such as software development, enterprise architecture or product lifecycle management. Thus, as M. Rosemann states [24], multinational corporations face the problem of managing collections of thousands of process models, and similarity measures are an important tool in finding the process model whose structure or behavior is closest to a specific problem. Similarities can therefore be used for the efficient management of large repositories, as J. Wang [30] states, since they allow us to find the appropriate process models among hundreds or thousands of possible candidates.

[☆]Supported by the Spanish government (cofinanced by FEDER funds) with the project TIN2015-65845-C03-02-R.

Email address: Valentin.Valero@uclm.es (Valentín Valero)

Petri nets are a useful formalism for describing and analyzing process models, because of their graphical nature and the solid mathematical foundations supporting them. Several timed extensions of the basic model have been proposed to expand their areas of application to those systems which exhibit time-dependent behavior that should be considered both in the modeling and the analysis process, such as distributed systems, communication systems and real-time systems. Timed business processes can therefore be modeled and analyzed with timed Petri nets.

Timed-Arc Petri Nets (TAPNs) [8, 16] are a timed extension of Petri nets in which tokens are assigned a natural value indicating the time elapsed from creation (their *age*), and arcs from places to transitions are labeled with time intervals that establish restrictions on the age of the tokens that can be used to fire the adjacent transitions. Thus, a transition can be fired when all its precondition places contain at least one token with an age in the time interval of the arc connecting the place with the transition. When the transition is fired, one token fulfilling this condition is removed from each precondition place, and one token of age 0 is produced in each postcondition place.

In this paper, two timed behavioral similarity measures over bounded Marked Labeled Timed-Arc Petri Nets (MLTAPNs) are introduced. They are called *btt-similarity* and *lcp-similarity*, and are defined by using a reduced version of the timed reachability graphs of the given MLTAPNs. The main drawback of this approach is the state space explosion, which is due to the enormous number of states these MLTAPNs will have in general. But the state graph size is not the only problem for the computation of timed behavioral similarities, since their computation is based on a comparison between the timed occurrence sequences of the given MLTAPNs, and obviously there will be, in general infinitely many of these sequences. The technique used in this paper is based on two timed versions of the longest common prefixes of the timed transition sequences of the MLTAPNs under consideration, but in contrast to other approaches loops are processed by considering a finite number of repetitions of them, which allows us to obtain a better approximation in repetitive behaviors.

This paper is structured as follows. The related work is presented in Section 2. Labeled Timed-Arc Petri Nets (TAPNs) and their semantics are formally defined in Section 3. Timed reachability graphs for marked TAPNs and a reduced version of them is defined in Section 4. The general concept of similarity and two timed similarity proposals are presented in Section 5. The evaluation of both similarities by experimental results is described in Section 6. Finally, the conclusions and plans for future work are presented in Section 7.

2. Related Work

A survey of the different approaches to introducing time into Petri nets was carried out by F.D.J. Bowden [9]. We can identify a first group of models, which assign time delays to transitions, either using a fixed and deterministic value [23] or choosing it from a probability distribution [3]. Other models use time intervals to establish the enabling times of transitions [21]. Finally, there are also

some models that introduce time on tokens [1, 8, 16]. In such a case, tokens are classified into two different classes: available and unavailable. Available tokens are those that can be immediately used for firing a transition, while unavailable tokens cannot. We have to wait for a certain period of time for these tokens to become available, although it is also possible for a token to remain unavailable forever (such tokens are said to be *dead*). Cerone and Maggiolo-Schettini [10] have defined a very general model (statically timed Petri nets), in which timing constraints are intervals that are statically associated with places, transitions and arcs. Thus, models with timing constraints attached only to places, transitions or arcs can be obtained as particular subclasses of this general framework.

In this paper we consider Timed-Arc Petri Nets (TAPNs) [8, 16], a timed extension of Petri nets in which tokens are assigned a natural value indicating the time elapsed from creation (their *age*), and arcs from places to transitions are also labeled with time intervals, which establish restrictions on the age of the tokens that can be used to fire the adjacent transitions. Timed-arc Petri nets are supported by some existing tools. TPAL [29] is a tool that allows us to obtain TAPN models from algebraic descriptions written in the TPAL language [28], and TAPAAL [12] is a tool that supports TAPNs extended with age invariants and both inhibitor and transport arcs, and which is also equipped with an efficient verification engine.

This is a rather pathological model, as reachability is undecidable [26], even in the case where tokens in different places are not required to age at the same rate [22], whereas some other known properties of Petri nets, such as boundedness, coverability and even termination are decidable [2, 13]. The detection of *dead transitions*, namely those that can never be fired, is also decidable [27] for this model.

As mentioned above, in this paper we define two timed behavioral similarity measures over bounded Labeled Timed-Arc Petri Nets (LTAPNs), in contrast to the structural approaches [14, 11, 20], which define measures on the basis of the model structure, disregarding its dynamic behavior. An intermediate approach has been taken by J. Bae et al. [6], where workflow dependency graphs are used to obtain normalized process network matrices, from which process similarity measures are computed as metric space distances between these normalized matrices. An untimed behavioral similarity measure based on transition adjacency relations in workflow nets has been defined in [32]. In this case, sequential firing of pairs of transitions is exploited in order to compute the TAR-similarity measure. Another untimed similarity measure has been defined by J. Wang et al. [30], in which principal transition sequences (PTS) are derived from the coverability tree of a Labeled Petri Net (LPN). These PTS are a characterization of the (possibly infinite) set of transition sequences of the LPN and are used to compute a similarity measure on the basis of their longest common subsequences. Z. Wang et al. [31] have also defined a similarity measure for untimed Petri nets by using the edit distance between node labeled versions of the coverability graphs. A different line of work exploits the observed behavior of the systems by using event logs: A.K. Alves de Medeiros et al.

[4] have defined a process trace equivalence measure based on the observed behavior, using event logs to capture their typical behavior. Finally, a survey of existing methods for performing a comparative graph analysis can be found in [15], in which the authors make a distinction between methods for deterministic and random graphs.

3. Labeled Timed-Arc Petri Nets

Labeled Timed-Arc Petri Nets (LTAPNs) have their transitions labeled with actions, their tokens are annotated with an age (an integer value indicating the time elapsed from creation), and the arcs connecting places with transitions are assigned a time interval which limits the age of the tokens that can be used to fire the adjacent transition.

However, a transition is not forced to fire when all its precondition places contain tokens with an adequate age, and the same is true even if the age of any of these tokens is about to expire. It is therefore possible that some tokens become *dead*, which means permanently useless because their increasing age will never allow the firing of any of their postcondition transitions.

Definition 1. (Labeled timed-arc Petri nets)

Let us consider an alphabet of actions Σ . A **labeled timed-arc Petri net** (LTAPN) is a tuple $N = (P, T, F, \mathbf{times}, \lambda)$, where P is a finite set of **places**, T is a finite set of **transitions** ($P \cap T = \emptyset$), F is the **flow relation** ($F \subseteq (P \times T) \cup (T \times P)$), \mathbf{times} is a function that associates to each arc (p, t) in F a pair of natural numbers, the second of which can be infinite, i.e.: $\mathbf{times} : F|_{P \times T} \rightarrow \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$, and $\lambda : T \rightarrow \Sigma$ is the labeling function that assigns an action to every transition. \square

When $\mathbf{times}(p, t) = [t_1, t_2]$ we write $\pi_i(p, t)$ to denote t_i , for $i = 1, 2$. We will always assume that $\pi_1(p, t) \leq \pi_2(p, t)$, and we will write $x \in \mathbf{times}(p, t)$ to denote $\pi_1(p, t) \leq x \leq \pi_2(p, t)$. We use the classical notation on Petri nets to denote the precondition and postcondition of both places and transitions:

$$\forall x \in P \cup T : \bullet x = \{y \mid (y, x) \in F\} \quad x^\bullet = \{y \mid (x, y) \in F\}$$

As mentioned above, tokens are annotated with natural numbers, so markings are defined by means of multisets on \mathbb{N} . More exactly, a marking M is a function $M : P \rightarrow \mathcal{B}(\mathbb{N})$ where $\mathcal{B}(\mathbb{N})$ denotes the set of finite multisets of natural numbers. Thus, as usual, each place is annotated with a certain number of tokens, but each one of them has a natural number (its *age*) associated with it. We will denote the set of markings of N by $\mathcal{M}(N)$, and using classical set notation, we denote the total number of tokens in a place p by $|M(p)|$.

As initial markings we only allow markings with all tokens of age zero¹. We define *marked labeled timed-arc Petri nets* (MLTAPNs) as pairs (N, M) , where N is an LTAPN and M is a marking on it.

¹It would not be a problem to permit initial markings with older tokens; the main reason for this restriction is to capture the intuitive idea that the system starts at time zero.

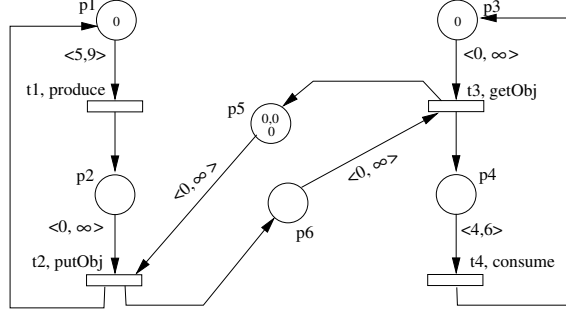


Figure 1: MLTAPN for the PC-problem.

A labeled timed-arc Petri net with an arbitrary marking can be graphically represented by extending the usual representation of P/T nets with the corresponding time information. In particular, we will use the age of each token to represent it. Therefore, MLTAPNs initially have a finite collection of zero values labeling each place. In Fig.1 we show an MLTAPN modeling a producer/consumer system, where the producer takes between 5 and 9 time units to produce an object, and the consumer takes between 4 and 6 time units to process an object. However, note that we are not forced to fire *produce* or *consume* transitions, even if the corresponding token ages are going to expire, so these tokens can become dead and the LTAPN will eventually become deadlocked in this case (only time elapsing is allowed when no transition can be fired).

Thus, an MLTAPN can evolve either by firing transitions or by time-elapsing. The following definition captures the dynamics of MTAPNs.

Definition 2. (Firing rule)

Let $N = (P, T, F, \mathbf{times}, \lambda)$ be an LTAPN, M a marking on it, and $t \in T$. We say that t is **enabled** at marking M if and only if:

$$\forall p \in \bullet t \exists x_p \in \mathbb{N} \text{ such that } M(p)(x_p) > 0 \wedge x_p \in \mathbf{times}(p, t),$$

i.e., on each precondition place of t we have some token whose age belongs to $\mathbf{times}(p, t)$. If t is enabled at M , it can be fired, and by its firing we reach a new marking M' , such that:

$$M'(p) = M(p) - C^-(p, t) + C^+(t, p), \forall p \in P$$

where both the subtraction and the addition operators work on multisets, $C^-(p, t)$ is defined for $p \in \bullet t$ in order to subtract one token from $M(p)$ with age $x_p \in \mathbf{times}(p, t)$, and it is empty for $p \notin \bullet t$. $C^+(t, p)$ is $\{0\}$ for $p \in t^\bullet$, otherwise it is empty.

Thus, from each precondition place p of t we remove one token whose age belongs to the time interval labeling the arc (p, t) , and we add a new token (with age 0) to each postcondition place of t .

As usual, we denote these evolutions by $M[t]M'$, but note that in general we will have several markings M' fulfilling this condition. Furthermore, the firing

of transitions does not consume any time. Therefore, time-elapsing is captured by the function **age**, which is defined below. By applying it, we age all the tokens of the net by the same time.

$$\mathbf{age} : \mathcal{M}(N) \times \mathbb{N} \longrightarrow \mathcal{M}(N)$$

$$\mathbf{age}(M, x)(p)(y) = \begin{cases} M(p)(y - x) & \text{if } y \geq x \\ 0 & \text{otherwise} \end{cases}$$

The marking obtained from M after x units of time without firing any transitions will be that given by $\mathbf{age}(M, x)$. \square

Note that we have introduced LTAPNs without weights on the arcs. It would not be a problem to introduce weights on them, with the usual interpretation. For PT-arcs (place to transition arcs) the weights on the arcs indicate the number of tokens that are required for the firing of the corresponding transition (all of them with ages in the associated time interval), and for TP-arcs (transition to place arcs) their weights indicate the number of new tokens that are produced in the corresponding postcondition places (all of them with age zero).

The semantics of an MLTAPN is defined in terms of timed transition sequences, which are defined by alternating time elapsing and the firing of transitions.

Definition 3. (Timed transition sequences)

Let $N = (P, T, F, \mathbf{times}, \lambda)$ be an LTAPN and M_0 a marking of it. A sequence $\sigma = M_0[t_1]_{x_1}M_1 \dots M_{n-1}[t_n]_{x_n}M_n$, where $M_0 \dots M_n$ are markings, $t_1 \dots t_n$ are transitions and $x_1 \dots x_n \in \mathbb{N}$, is a *timed transition sequence* of N if and only if $\mathbf{age}(M_{i-1}, x_i)[t_i]M_i$, for $i = 1, \dots, n$. Note that in this definition we permit $x_i = 0$ in order to allow the firing of several transitions at the same instant. The sequences obtained by replacing the transitions with their associated actions are here called *timed occurrence sequences*.

We then say that M_n is a marking reachable from M_0 in time $x = \sum_{i=1}^n x_i$, denoted by $M \in [(N, M_0)]_x$ (or simply $[M_0]_x$ when the underlying LTAPN is clear from the context). The set of all reachable markings of N starting from M_0 is denoted by $[(N, M_0)]$ (or simply $[M_0]$), and the set of all reachable markings up to time x is denoted by $[(N, M_0)]_{\leq x}$. \square

MLTAPNs are said to be 1-safe when for every reachable marking we have at most one token in every place. A place p is said to be bounded if there exists $n \in \mathbb{N}$ such that for all $M \in [M_0]$ we have $|M(p)| \leq n$. Then, an MLTAPN is bounded if all its places are bounded. We also define for every $p \in P$ the values $Max_1(p) = Max\{\pi_1(p, t) \mid t \in p^\bullet\}$, $Max_2(p) = Max\{\pi_2(p, t) \mid t \in p^\bullet, \pi_2(p, t) < \infty\}$, $Max(p) = Max\{Max_1(p), Max_2(p)\}$, and $S(p) = 1 + Max(p)$. Note that when the age of a token in a place p is greater than $S(p)$, it can only be used to fire transitions for which $\pi_2(p, t) = \infty$, i.e., the specific age of the token is unimportant once it exceeds $S(p)$. Thus, any token in p with an age greater than $S(p)$ can be considered as having age $S(p)$, since this will not alter

the behavior of the net. Thus, from now onwards we will assume that $S(p)$ is the maximum age for the tokens in p and $S \in \mathbb{N}^{|P|}$ will denote the vector with these values.

We will also consider $\pi_2^*(p, t)$, which is defined as follows:

$$\forall p \in P, \forall t \in p^\bullet : \pi_2^*(p, t) = \begin{cases} \pi_2(p, t) & \text{if } \pi_2(p, t) < \infty \\ S(p) & \text{otherwise} \end{cases}$$

Theorem 1. Let (N, M_0) be an MLTAPN and N^* the LTAPN obtained from N by replacing π_2 with π_2^* . Let $age^*(M, x)$ be the modified *age* function that freezes the age of tokens at the maximum value $S(p)$.

$$\forall x \in \mathbb{N}, \forall p \in P, \forall y \in [0, S(p)], \forall M \in \mathcal{M}(N):$$

$$age^*(M, x)(p)(y) = \begin{cases} \sum_{\{x' \in \mathbb{Z} \mid x' \leq x, y - x' \geq S(p)\}} M(p)(y - x') & \text{if } y = S(p) \\ M(p)(y - x) & \text{if } y < S(p), \text{ and} \\ & S(p) > y - x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Then, $M_0[t_1]_{x_1} M_1 \dots M_{n-1}[t_n]_{x_n} M_n$ is a timed transition sequence of (N, M_0) if and only if $M'_0[t_1]_{x_1} M'_1 \dots M'_{n-1}[t_n]_{x_n} M'_n$ is a timed transition sequence of (N^*, M'_0) with the modified firing rule, where $M'_0 = age^*(M_0, 0)$, and $age^*(M'_{i-1}, x_i)[t_i] M'_i$ for $i = 1, \dots, n$.

Proof: Every marking M'_i is obtained from the corresponding M_i by adjusting the token ages greater than $S(p)$ to exactly this value (for all p). Specifically, $age^*(M_0, 0)$ takes all the tokens with an age greater than $S(p)$ (for all places p) as having age $S(p)$. Since $S(p)$ is greater than any (finite) value in the time intervals of the outgoing arcs from p , all transitions enabled at M_i are also enabled at M'_i (and viceversa), and the marking M_{i+1} obtained with the firing of t_{i+1} corresponds to M'_{i+1} . \square

With this theorem we have established the equivalence between the given MLTAPN and the modified one, which we call MLTAPN* from now onwards.

4. Timed Reachability Graph

We now restrict our attention to bounded LTAPNs, for which in every reachable marking the number of tokens in every place is bounded by some natural n . For these bounded LTAPNs we can construct a finite reachability graph, considering the aforementioned modified firing rule, in which ages greater than $S(p)$ in every place p are frozen at this value. However, we know that even for untimed bounded Petri nets the size of the reachability graph is intractable in general, so the problem becomes worse when ages are introduced, and the practical use of this construction is then limited by these restrictions. In this graph we have

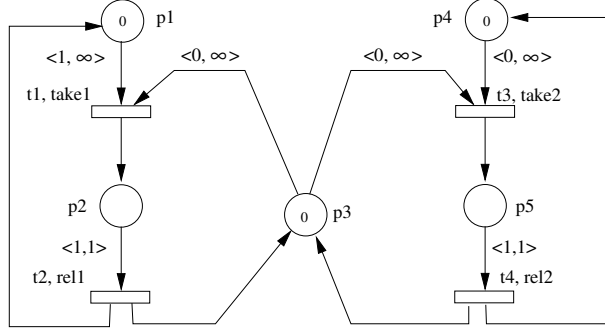


Figure 2: MLTAPN for the mutual exclusion problem.

two types of edges: those capturing the transition firings and those capturing time elapsing. The former are labeled with transitions (actions), and the latter with a time delay of one time unit. Note, however, that we can have several edges leaving a node labeled with the same transition, because the model is non-deterministic and we can have several tokens with different ages allowing the firing of a transition.

Definition 4. (Timed reachability graph)

We define the timed reachability graph of a bounded MLTAPN* (N, M_0) , denoted by $\mathcal{G}(N, M_0)$, as the triple $([M_0], M_0, \mapsto)$, where $\mapsto = \{(M, t, M') \mid M \in [M_0], M[t]M', t \in T\} \cup \{(M, 1, M') \mid M \in [M_0], M' = age^*(M, 1)\}$. \square

Figure 2 shows an MLTAPN modeling the mutual exclusion problem. Its corresponding timed reachability graph, which is depicted in Figure 3, has 15 nodes (markings). As this MLTAPN is 1-safe, markings in the graph are represented by indicating the corresponding token ages in the place indexes or by a dash when there is no token in that place. The timed reachability graph of an MLTAPN can be quite large in general, which is an obvious consequence of the many different combinations of token ages that can be obtained as the system evolves in many different ways. For instance, for the PC problem shown in Figure 1 we have obtained, with the TPAL tool [29], a timed reachability graph with 428 nodes, with $S = (10, 1, 1, 7, 1, 1)$, in contrast with the 16 un-timed markings that are reachable for the un-timed version of the PC problem. We can reduce the graph size slightly by joining several consecutive delay transitions into a single one, i.e., when we find several consecutive delays and we do not have any enabled transitions in the intermediate markings, we can join the consecutive delays into a single delay transition labeled with the cumulative time delay.

Definition 5. (Reduced timed reachability graph)

We define the reduced timed reachability graph of a bounded MLTAPN (N, M_0) , denoted by $\mathcal{R}(N, M_0)$, as the triple (Q, M_0, \rightarrow) , where $Q \subseteq [M_0]$, $M_0 \in Q$ and the other markings in Q are those obtained by the application of the relation \rightarrow , defined as follows:

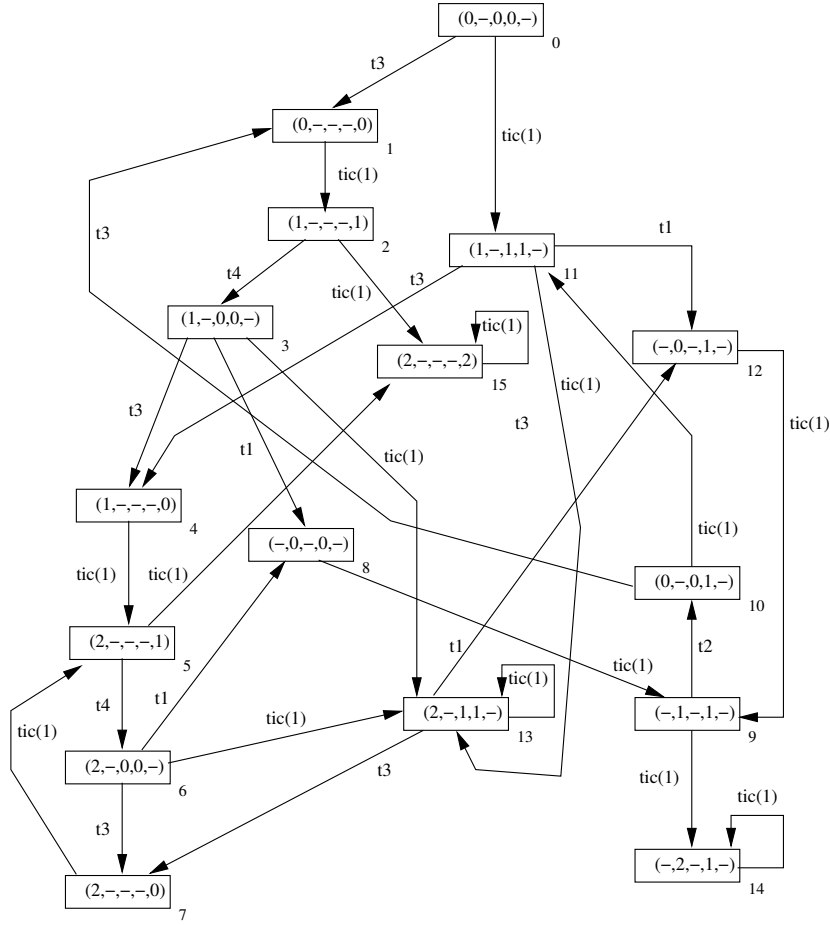


Figure 3: Timed reachability graph for the MLTAPN in Figure 2.

- $(M, t, M') \in \mapsto \Rightarrow (M, t, M') \in \rightarrow$ and $M, M' \in Q$.
- $\forall M_1, \dots, M_n$ such that $M_{i+1} = \text{age}^*(M_i, 1)$ ($i = 1, \dots, n - 1$) and such that $\exists t_n \in T, M_n[t_n] \wedge \forall i = 1, \dots, n - 1$ there is no transition $t_i \in T, M_i[t_i]$, then $(M_1, n, M_n) \in \rightarrow$ and $M_1, M_n \in Q$.

□

In this definition Q is the set of markings that are obtained as a result of the construction, which is most likely to be a strict subset of $[M_0)$, but it is important to observe that we are not losing any information about the reachable markings in the given MLTAPN, because each delay transition (M, x, M') , with $x > 1$ also represents every intermediate marking reachable from M by a delay of $x' < x$ time units. For instance, for the PC problem in Fig. 1 we have obtained, with the TPAL tool, a reduced timed reachability graph of 364 nodes (64 nodes less than the whole graph).

Let us call *action edges* those edges of $\mathcal{R}(N, M_0)$ that correspond to the firing of transitions, and *delay edges* those edges that correspond to time elapsing.

Proposition 1. Let (N, M_0) be an MLTAPN and $\mathcal{R}(N, M_0)$ its corresponding reduced reachability graph. For every node in $\mathcal{R}(N, M_0)$ there can be at most one delay edge leaving the node.

Proof: Immediate, because this property obviously holds for the timed reachability graph, and the reduced graph is obtained from that graph by collapsing a sequence of delay edges into a single edge when no action edge appears at the intermediate nodes. \square

Another reduction that can sometimes also be applied is that related to time escalation, with the goal of reducing the number of timed reachable markings by dividing the numbers in the time intervals by their greatest common divisor, when this value is greater than one. The following theorem states that in these conditions every reachable marking M in an MLTAPN (N, M_0) , with all the token ages in M either zero or divisible by g , can also be reached in the MLTAPN N' constructed by applying the age stabilization and time escalation, and viceversa.

Theorem 2. Let (N, M_0) be an MLTAPN, and $g = \gcd\{\pi_i(t) \mid 0 < \pi_i(t) < \infty, t \in T\} > 1$. Let $St(N) = \text{Max}\{S(p) \mid p \in P\}$, and $St'(N) = \text{min}\{n \in \mathbb{N}^+ \mid n \geq St(N), n \bmod g = 0\}$, and (N^*, M_0) the MLTAPN* obtained from (N, M_0) taking $St'(N)$ as the maximum token age for all the places.

Let us now consider N' as the MLTAPN* constructed from N^* , with the same structure, but replacing π with π' , which is defined as follows: $\forall t \in T, \pi'_i(t) = \pi_i(t)/g$, for $i = 1, 2$. Then, for all $x \in \mathbb{N}$ divisible by g and every marking $M \in \mathcal{M}(N)$ with all its tokens with age either zero or a value divisible by g , we have $M \in [(N, M_0)]_x$ if and only if $M/g \in [(N', M_0/g)]_{x/g}$, where M/g is defined as follows: $\forall p \in P, \forall y \in \mathbb{N}, M/g(p)(y) = M(p)(y \cdot g)$.

Proof: The left implication is immediate, so let us consider the direct implication.

Let $\sigma = M_0[t_1]_{x_1} M_1 \dots M_{n-1}[t_n]_{x_n} M_n = M$ be the timed transition sequence leading to M , with $x = \sum_{i=1}^n x_i$.

We call $\alpha_i = \sum_{j=1}^i x_j$, which are the instants at which transitions t_i are fired. We then consider the instants $\alpha'_i = k_i \cdot g$, with $k_i \in \mathbb{N}$ such that $-g/2 \leq \alpha'_i - \alpha_i < g/2$ (Eq.1), i.e., the closest instants to α_i that are divisible by g . Observe that $\alpha'_n = x = \alpha_n$, since x is divisible by g . Eq.1 can then be rewritten as $-g/2 \leq k_i \cdot g - \sum_{j=1}^i x_j < g/2$ (Eq.2).

We then move the firing of each transition t_i to the instant α'_i , and we call M'_i the marking obtained as a result of this firing. We should see that this firing is actually possible at that instant α'_i . We use induction on i for this proof, starting from $M_0 = M'_0$, and showing that at each step M'_i only differs from M_i in the ages of the tokens, i.e., they have the same number of tokens in each place, but with different ages, whose difference is actually bounded as follows:

$$-g/2 < e_i^p - e_i'^p \leq g/2 \quad (\text{eq.3})$$

for the same token in place p with age e_i^p in M_i and $e_i'^p$ in M_i' , which is trivially true for $i = 0$.

We now take $i > 0$, assuming that Eq.3 is true for $i - 1$, and we know that transition t_i is fireable at M_{i-1} at the instant α_i , so we must have $t_{i_1}^p \leq e_{i-1}^p + x_i \leq t_{i_2}^p$ (Eq.4), where $\text{times}(p, t) = (t_{i_1}^p, t_{i_2}^p)$.

Our goal, therefore, is to prove $M_{i-1}'[t_i]x_i'M_i'$, where $x_i' = \alpha_i' - \alpha_{i-1}'$. We first see that $|x_i - x_i'| < g$. Using the definition of x_i' and Eq.2 for i and $i - 1$ we obtain the following chain of inequalities:

$$x_i' - x_i = (k_i - k_{i-1})g - x_i < g/2 + \sum_{j=1}^i x_j - k_{i-1}g - x_i = g/2 + \sum_{j=1}^{i-1} x_j - k_{i-1}g \leq g/2 + g/2 = g.$$

Taking Eq.2 again we can write the inequalities:

$$x_i' - x_i = (k_i - k_{i-1})g - x_i \geq -g/2 + \sum_{j=1}^i x_j - k_{i-1}g - x_i = -g/2 + \sum_{j=1}^{i-1} x_j - k_{i-1}g > -g/2 - g/2 = -g.$$

Thus, we conclude that $|x_i - x_i'| < g$.

Let us now take every place $p \in \bullet t_i$. We must prove that $t_{i_1}^p \leq e_{i-1}'^p + x_i' \leq t_{i_2}^p$, and thus t_i will be enabled at M_{i-1}' at the instant α_i' .

- Using Eq.3 for $i - 1$ and Eq.4 we have $e_{i-1}'^p + x_i' < e_i^p + g/2 + x_i' \leq t_{i_2}^p + g/2 + (x_i' - x_i)$. Now using $|x_i - x_i'| < g$, we conclude $e_{i-1}'^p + x_i' < t_{i_2}^p + 3g/2$. Taking into account that $e_{i-1}'^p, x_i'$ and $t_{i_2}^p$ are all divisible by g , we must have $e_{i-1}'^p + x_i' \leq t_{i_2}^p$.
- Again using Eq.3 for $i - 1$ and Eq.4 we have $e_{i-1}'^p + x_i' \geq e_{i-1}^p - g/2 + x_i' \geq t_{i_1}^p - x_i - g/2 + x_i' > t_{i_1}^p - g - g/2 = t_{i_1}^p - 3g/2$. Once again, $e_{i-1}'^p, x_i'$ and $t_{i_1}^p$ are all divisible by g , so we conclude $e_{i-1}'^p + x_i' \geq t_{i_1}^p$.

Thus, t_i can be fired at the marking M_{i-1}' , and the new marking obtained from this firing is M_i' , which only differs from M_i in the token ages, as we have modified the instant at which t_i fires in $|\alpha_i - \alpha_i'|$ time units. Thus, from Eq.2 we immediately conclude $-g/2 < e_i^p - e_i'^p \leq g/2$. \square

5. Similarity Between MLTAPNs

In the previous section a reduced timed reachability graph is defined for bounded MLTAPNs. We now use this finite graph to define a timed similarity measure between two given bounded MLTAPNs. The state space explosion problem is obviously an impediment for the practical application of this technique, not only due to the graph size, but mainly because of the cost of the similarity computation. Therefore, in this paper we only present the theoretical concepts and their applications to simple examples. Other possible approaches for dealing with this problem, such as approximation techniques based on testing can be considered for those cases in which the problem becomes intractable, and this will be the focus of further research in this field.

The behavior of an MLTAPN is captured by all the timed transition sequences we can obtain from its initial marking, which are obtained by considering all the possible paths from the root node of $\mathcal{R}(N, M_0)$. However, even if the

given MLTAPN is bounded, we will usually have infinitely many sequences, due to marking repetitions (loops). Thus, some approaches consider a (partial) finite characterization. For instance, J. Wang [30] defines the so-called Principal Transition Sequences (PTS) for untimed Petri nets, by using the coverability tree. Principal transition sequences divide the transition sequences into three parts: the terminal transition sequences (from the initial marking to dead markings), the prefix transition sequences for loop behaviors (from the initial marking to a repeated marking), and the loops (from a repeated marking to the leaf node labeled with this same marking), respectively called $\mathcal{P}_1, \mathcal{P}_2$ and \mathcal{P}_3 . For the Petri nets shown in Figure 4 we easily obtain $\mathcal{P}_1 = \emptyset, \mathcal{P}_2 = \{a\}$ and $\mathcal{P}_3 = \{a\}$ for the PN on the left-hand side, and $\mathcal{P}'_1 = \emptyset, \mathcal{P}'_2 = \emptyset$ and $\mathcal{P}'_3 = \{a\}$ for the PN on the right-hand side. The coverability trees of both PNs are depicted in the figure at their side ($T(N)$ and $T(N')$). The similarity notion is defined on a pairwise comparison of these sets, on the basis of the longest common subsequences in the related sets. Thus, $Sim(\mathcal{P}_1, \mathcal{P}'_1) = 1, Sim(\mathcal{P}_2, \mathcal{P}'_2) = 0$ and $Sim(\mathcal{P}_3, \mathcal{P}'_3) = 1$. A weight is also used to take into account the number of elements in these sets. In this case the values obtained are $\lambda_1 = 0, \lambda_2 = 1/3, \lambda_3 = 2/3$. Thus, the similarity between N and N' is finally $2/3$, which does not even change if we replace the label of t_1 with any other action. We can therefore conclude that this similarity concept does not deal properly with loops, since both Petri nets, N and N' , have the same observable behavior.

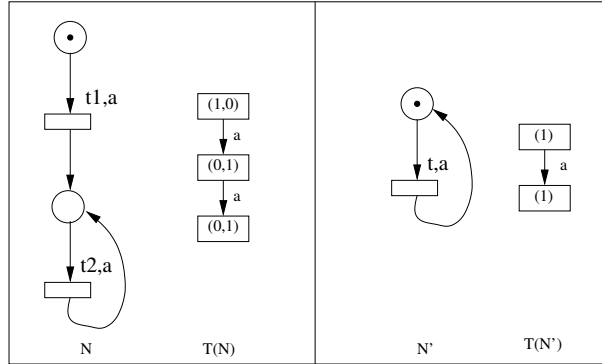


Figure 4: Petri Nets with the same observable behavior.

In order to improve the similarity measure in situations such as the one illustrated in Figure 4 we take a different approach, and this consists in using bounded loop timed occurrence sequences, in which we restrict the number of iterations in the loops by a certain constant n . Thus, we obtain a finite set of n -bounded loop occurrence transition sequences.

Definition 6. (Bounded loop timed transition sequences)

Let (N, M_0) be a bounded MLTAPN, $n \in \mathbb{N}$ and $\mathcal{R}(N, M_0)$ the reduced timed reachability graph of (N, M_0) . In this definition, we extend the function λ to

act both on transitions and natural numbers, as follows:

$$\lambda(u) = \begin{cases} \lambda(t) & \text{if } u \in T \\ i & \text{if } u = i \in \mathbb{N} \end{cases}$$

We define the n -bounded loop timed occurrence sequences of (N, M_0) as all the maximal paths we can construct from the root by performing the loops in at most n times:

$$BTTS_n(N, M_0) = \{ \lambda(u_0) \dots \lambda(u_{m-1}) \mid \sigma = q_0 \xrightarrow{u_0} q_1 \dots q_{m-1} \xrightarrow{u_{m-1}} q_m \text{ is a} \\ \text{maximal path of } \mathcal{R}(N, M_0) \text{ with } q_0 = M_0, m \in \mathbb{N}^+, u_i \in T \cup \mathbb{N}^+, \\ \#(q_i, \sigma) \leq n, \text{ for all } i = 1, \dots, m \}$$

where $\#(q, \sigma)$ denotes the number of repetitions of marking q in the sequence σ , and a path σ in the graph is considered to be maximal if it cannot be extended to $\sigma' = \sigma \xrightarrow{u} q$ with $\#(q, \sigma') \leq n$. \square

Note that we consider sequences of both actions and delays in this definition, no matter the intermediate markings that are traversed in the corresponding path. In fact, the same sequence can be obtained by taking several different paths through the graph, so we are actually defining a multiset of timed occurrence sequences. From a practical point of view it would be better to define a set and check the similarity of each one of the possible sequences with the behavior of the other MLTAPN, but $BTTS_n$ must be computed *on-the-fly* due to its potentially enormous size, so we cannot easily detect whether a sequence was obtained previously or not.

Furthermore, we consider that the sequences in $BTTS_n(N, M_0)$ do not have two or more consecutive time transitions, since we can consider the equivalent sequence in which all the consecutive time transitions are accumulated, i.e. $a.i_1.i_2.\dots.i_r.b$ would be replaced by $a.\sum_{j=1}^r i_j.b$. Actually, we consider all the sequences following the syntax $\Sigma_t^* = \mathbb{N} \times (\Sigma \times \mathbb{N})^*$, i.e. a (possibly null) initial delay is followed by a sequence of alternating actions and (possibly null) delays. Thus, a normalization is applied to the sequences obtained from the reduced timed reachability graph in order to obtain these *normalized sequences*. Therefore, in the following we will always consider normalized sequences.

For instance, for $a.b.7.c$ we take the equivalent sequence $0.a.0.b.7.c.0$ as representative.

Proposition 2. Let (N, M_0) be a bounded MLTAPN and $n \in \mathbb{N}^+$. $BTTS_n(N, M_0)$ is finite.

Proof: Immediate, since $\mathcal{R}(N, M_0)$ is finite, and every state can be repeated at most n times. \square

As an illustration, for the mutual exclusion problem shown in Figure 2 we have obtained the results shown in Figure 5, in which we show the size of $BTTS_n(N, M_0)$ for $n = 1, 2, 3, 4$. In this figure we see the exponential increase

n	$ BTTS_n $
1	56
2	2786
3	127088
4	5749802

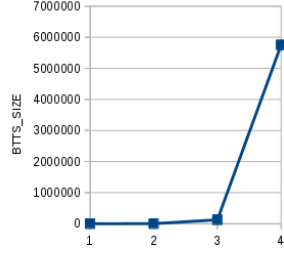


Figure 5: $BTTS_n$ sizes for the Mutual Exclusion Problem.

in the size of $BTTS_n$, but we do not need to take high values for n , since MLTAPN's behavior is in general well captured by just taking the low values ($n = 1, 2, 3$).

Let us now study how we can define a *similarity* function on MLTAPNs that use the same alphabet Σ . Let us assume first that we have a similarity function s that maps a pair of timed occurrence sequences to a real number between 0 and 1, $s : \Sigma_t^* \times \Sigma_t^* \rightarrow [0, 1]$, which fulfills the following properties for all pairs of sequences σ_1, σ_2 :

1. $\sigma_1 = \sigma_2 \Leftrightarrow s(\sigma_1, \sigma_2) = 1$.
2. $s(\sigma_1, \sigma_2) = s(\sigma_2, \sigma_1)$.

These properties state that s is symmetric and it can only be 1 for two exactly equal sequences, otherwise it will always return a value smaller than 1 ($1 - s$ is a semi-metric). We now assume that we have a *selector function* $\psi(\mathcal{R}_1, \mathcal{R}_2, \sigma_1) = \sigma_2$, where $\mathcal{R}_1, \mathcal{R}_2$ are reduced timed reachability graphs from two possibly different MLTAPNs and σ_1 is a timed occurrence sequence in \mathcal{R}_1 , which provides us with a timed occurrence sequence σ_2 in \mathcal{R}_2 , such that $\psi(\mathcal{R}, \mathcal{R}, \sigma) = \sigma$, for all \mathcal{R} and σ .

Definition 7. (Similarity of MLTAPNs)

Let $(N_1, M_{0,1}), (N_2, M_{0,2})$ be two bounded MLTAPNs, $n \in \mathbb{N}^+$, ψ a selector function and s a similarity function over timed occurrence sequences in Σ_t^* . Let us call $B_i = BTTS_n(N_i, M_{0,i})$, $|B_i|$ the size of B_i as a multiset ($i = 1, 2$) and $\psi_{ij}(\sigma) = \psi(\mathcal{R}_i, \mathcal{R}_j, \sigma)$, $i, j = 1, 2$, where \mathcal{R}_i ($i = 1, 2$) is the reduced reachability graph of $(N, M_{0,i})$. We define

$$Sim_{\psi, s, n}((N_1, M_{0,1}), (N_2, M_{0,2})) = \frac{\sum_{\sigma_1 \in B_1} s(\sigma_1, \psi_{12}(\sigma_1)) + \sum_{\sigma_2 \in B_2} s(\psi_{21}(\sigma_2), \sigma_2)}{|B_1| + |B_2|}$$

□

Proposition 3. $Sim_{\psi, s, n}$ is well-defined and fulfills the properties:

1. $Sim_{\psi, s, n}((N, M_0), (N, M_0)) = 1$.
2. $Sim_{\psi, s, n}((N_1, M_{0,1}), (N_2, M_{0,2})) = Sim_{\psi, s, n}((N_2, M_{0,2}), (N_1, M_{0,1}))$.

Proof: From Proposition 2 we know that B_i are finite, and they always contain at least one element because time elapsing is always possible, so $|B_i| > 0$, $i = 1, 2$. Hence it is well defined. Property 1 is immediate, because by the definition of ψ we know $\psi(\mathcal{R}, \mathcal{R}, \sigma) = \sigma$ and $s(\sigma, \sigma) = 1$. Property 2 follows immediately from the symmetry of s . \square

This definition computes the similarity of every sequence in B_1 (resp. B_2) with its counterpart in B_2 (resp. B_1) obtained by applying the selector function ψ . All these values are summed and the resulting value is divided by the total number of sequences. The resulting value is thus the global average of the similarities that have been computed.

Thus, we have to select both a selector function and a similarity function for timed occurrence sequences. The idea behind the selector function is that the sequence σ_2 should have the maximum similarity with σ_1 , so as to obtain a maximum value for the similarity of both MLTAPNs. A first definition can therefore take σ_2 as being the sequence in $BTT S_n(N_2, M_{0,2})$ yielding the maximum value of $s(\sigma_1, \sigma_2)$.

Definition 8. (btt-selector function) Let $(N_1, M_{0,1}), (N_2, M_{0,2})$ be two bounded MLTAPNs, s a similarity function for timed occurrence sequences and $n \in \mathbb{N}^+$. We define the *btt-selector function* ψ_b^s as follows:

$\forall \sigma_1 \in BTT S_n(N_1, M_{0,1}) : \psi_b^s(\sigma_1) = \sigma_2 \in BTT S_n(N_2, M_{0,2})$, such that $\forall \sigma'_2 \in BTT S_n(N_2, M_{0,2})$ we have $s(\sigma_1, \sigma'_2) \leq s(\sigma_1, \sigma_2)$. \square

It is immediate to see that ψ_b^s is well defined, but it has the problem of choosing σ_2 in a restricted multiset ($BTT S_n(N_2, M_{0,2})$), which can be quite different from $BTT S_n(N_1, M_{0,1})$, because both MLTAPNs can have quite a significantly different number of states in their graphs. Figure 6 shows two MLTAPNs that intuitively have the same behavior. In the second column of this figure their corresponding reduced reachability graphs are depicted, and their normalized $BTT S_n$ are shown in the third column, for $n = 1, 2$. We can observe that for $n = 1$ the only sequence in $BTT S_1^1$ is a single tic $\{1\}$, whereas for N_2 we also have $0.a.1$. Thus, intuitively, sequence 1 matches in both sets, but we do not have a good sequence in $BTT S_1^1$ matching sequence $0.a.1$ in $BTT S_1^2$. Observe that this problem is not solved by taking greater values of n , because both (multi)sets $BTT S_n^1$ and $BTT S_n^2$ are still quite different.

Therefore, we now introduce the *lcp-selector function*, which takes the timed occurrence sequence σ_2 from $\mathcal{R}(N_2, M_{0,2})$ that has the longest common prefix with σ_1 .

Definition 9. (lcp-selector function) Let $(N_1, M_{0,1}), (N_2, M_{0,2})$ be two bounded MLTAPNs, s a similarity function for timed occurrence sequences and $n \in \mathbb{N}^+$. We define the *lcp-selector function* ψ_p^s as follows:

$\forall \sigma_1 \in BTT S_n(N_1, M_{0,1}) : \psi_p^s(\sigma_1) = \sigma_2$, such that σ_2 is a path of $\mathcal{R}(N_2, M_{0,2})$ and it is either a maximal prefix of σ_1 or they only differ in the last element (a delay). \square

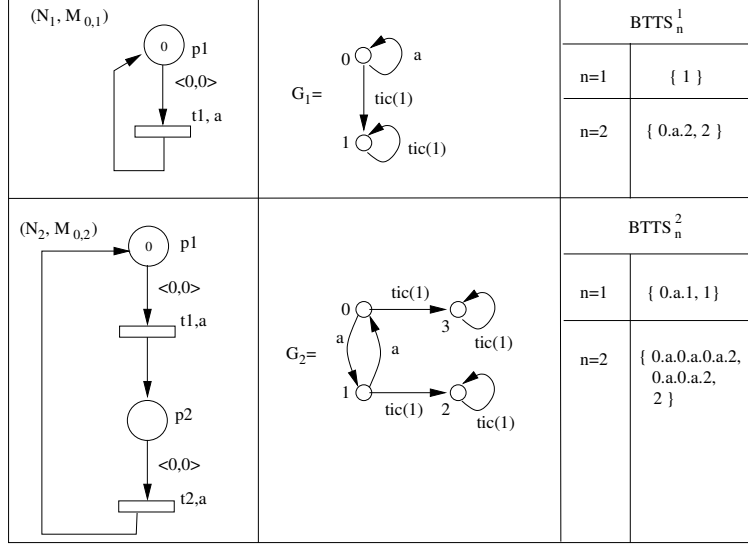


Figure 6: Two MLTAPNs with the same behavior.

According to this definition, when both sequences σ_1 and σ_2 coincide, except possibly in the final delay, we can take $\psi_p^s(\sigma_1) = \sigma_2$, because we assume that by taking the minimum of both delays we can obtain a common sequence. Obviously, we could have several σ_2 fulfilling this condition by just taking different final delays for σ_2 , but we can select any of them for our purposes.

For the example in Figure 6, now taking the $BTTS_n$ -sequences σ_1 of $(N_2, M_{0,2})$, we have $\psi_p^s(1) = 1$ and for $\sigma_1 = 0.a.1 \in BTTS_1^2$ we have $\psi_p^s(0.a.1) = 0.a.1$, since $0.a.1$ is a (normalized) timed occurrence sequence that can be obtained from G_1 and obviously it has the longest common prefix with $0.a.1$.

We now face the problem of choosing a suitable similarity function for timed occurrence sequences, capturing the intuition behind a *timed similar behavior*. Taking as reference timed bisimulations, in which any action or delay in one sequence must have its corresponding counterpart in the other sequence, we can consider as being more similar those sequences that match further in the comparison, but also taking into account the length of both sequences, in the same line as in [30], but we must now deal with time delays between actions, and weight their impact on the similarity.

In this paper we take a *strong approach*, in the sense that we consider time as relevant as actions, so the similarity function stops at the first mismatching either in actions or delays, so it is exactly based on the longest common prefix subsequence. This similarity function could be used in hard real-time systems, in which time restrictions must be strictly fulfilled.

Definition 10. (Timed strong similarity)

For every pair $\sigma_1, \sigma_2 \in \Sigma_t^*$ we define the strong similarity between σ_1 and σ_2 as

follows:

$$s_0(\sigma_1, \sigma_2) = \frac{|lcp_s(\sigma_1, \sigma_2)|}{\max\{|\sigma_1|, |\sigma_2|\}}$$

where $|\sigma_i|$ stands for the length of σ_i and $|lcp_s(\sigma_1, \sigma_2)|$ stands for the length of the longest common prefix subsequence, without distinguishing between actions and delays, with the exception of the final delay if they only differ in this element, in which case they are considered as having the same length, including this final delay element. \square

It follows immediately that s_0 fulfills the properties indicated above for valid similarities.

Example 1. For the sequences below we obtain the indicated strong similarities:

1. For $\sigma_{1,1} = 0.a.1.b.0$ and $\sigma_{1,2} = 1.a.1.b.0$, we have $s_0(\sigma_{1,1}, \sigma_{1,2}) = 0$.
2. For $\sigma_{2,1} = 0.a.1.b.0$ and $\sigma_{2,2} = 0.a.2.b.0$, we have $s_0(\sigma_{2,1}, \sigma_{2,2}) = 2/5$.
3. For $\sigma_{3,1} = 1.a.1.b.0$ and $\sigma_{3,2} = 1.a.1.c.0.f.1.d.3$, we have $s_0(\sigma_{3,1}, \sigma_{3,2}) = 3/9$.
4. For $\sigma_{4,1} = 1.a.1$ and $\sigma_{4,2} = 1.a.3$, we have $s_0(\sigma_{4,1}, \sigma_{4,2}) = 1$. \square

Definition 11. (Timed strong distance)

For every pair $\sigma_1, \sigma_2 \in \Sigma_i^*$ we define the *timed strong distance* between σ_1 and σ_2 as follows: $d_0(\sigma_1, \sigma_2) = 1 - s_0(\sigma_1, \sigma_2)$. \square

Proposition 4. The *timed strong distance* is a metric.

Proof: s_0 is a semi-metric, so we only need to prove that the triangular inequality holds for d_0 . Let us consider $\sigma_1, \sigma_2, \sigma_3$, with lengths n_1, n_2, n_3 , and $L_{i,j} = |lcp_s(\sigma_i, \sigma_j)|$, for $i, j = 1, 2, 3, i \neq j$.

We must prove

$$d_0(\sigma_1, \sigma_2) \leq d_0(\sigma_1, \sigma_3) + d_0(\sigma_3, \sigma_2) \Leftrightarrow$$

$$1 - \frac{L_{1,2}}{\max\{n_1, n_2\}} \leq 1 - \frac{L_{1,3}}{\max\{n_1, n_3\}} + 1 - \frac{L_{2,3}}{\max\{n_2, n_3\}}$$

For symmetry, we can assume $n_1 \leq n_2$. We can now distinguish three cases:

1. If $n_3 \leq n_1$, the inequality above is rewritten as follows:

$$-\frac{L_{1,2}}{n_2} \leq 1 - \frac{L_{1,3}}{n_1} - \frac{L_{2,3}}{n_2} \Leftrightarrow L_{1,3}.n_2 + L_{2,3}.n_1 \leq n_1.n_2 + L_{1,2}.n_1$$

which is equivalent to $(L_{2,3} - L_{1,2}).n_1 \leq (n_1 - L_{1,3}).n_2$.

If $L_{2,3} \leq L_{1,2}$, the result is immediate, because $L_{1,3} \leq n_1$. Otherwise, if $L_{2,3} > L_{1,2}$, it follows that $L_{1,3} = L_{1,2}$. Then, from $n_1 \leq n_2$ and $n_3 \leq n_1$, we have:

$$(L_{2,3} - L_{1,2}).n_1 \leq (L_{2,3} - L_{1,2}).n_2 \leq (n_1 - L_{1,3}).n_2$$

which finishes the proof.

2. If $n_1 \leq n_3 \leq n_2$, the inequality above is now rewritten as follows:

$$-\frac{L_{1,2}}{n_2} \leq 1 - \frac{L_{1,3}}{n_3} - \frac{L_{2,3}}{n_2} \Leftrightarrow L_{1,3}.n_2 + L_{2,3}.n_3 \leq n_2.n_3 + L_{1,2}.n_3$$

which is equivalent to $(L_{2,3} - L_{1,2}).n_3 \leq (n_3 - L_{1,3}).n_2$.

If $L_{2,3} \leq L_{1,2}$, the result is immediate, because $L_{1,3} \leq n_3$. Otherwise, if $L_{2,3} > L_{1,2}$, it follows that $L_{1,3} = L_{1,2}$. Then, from $n_3 \leq n_2$ and $L_{2,3} \leq n_3$, we have:

$$(L_{2,3} - L_{1,2}).n_3 \leq (L_{2,3} - L_{1,2}).n_2 \leq (n_3 - L_{1,3}).n_2$$

which finishes the proof.

3. If $n_1 \leq n_2 \leq n_3$, the inequality above is now rewritten as follows:

$$-\frac{L_{1,2}}{n_2} \leq 1 - \frac{L_{1,3}}{n_3} - \frac{L_{2,3}}{n_3} \Leftrightarrow L_{1,3}.n_2 + L_{2,3}.n_2 \leq n_2.n_3 + L_{1,2}.n_3$$

If $L_{2,3} \leq L_{1,2}$, the result is immediate, because $L_{1,3} \leq n_3$ and $n_2 \leq n_3$. Otherwise, if $L_{2,3} > L_{1,2}$, it follows that $L_{1,3} = L_{1,2}$. Then, we have

$$L_{1,3}.n_2 + L_{2,3}.n_2 = L_{1,2}.n_2 + L_{2,3}.n_2 \leq L_{1,2}.n_3 + L_{2,3}.n_2 \leq L_{1,2}.n_3 + n_3.n_2$$

which finishes the proof. \square

Definition 12. (Strong similarities)

We define the following strong similarities on MLTAPNs, according to Def. 7:

1. $Sim_{\psi_b^{s_0}, s_0, n}$ is the similarity obtained using the timed strong similarity s_0 and the btts-selector function $\psi_b^{s_0}$ (*btts-similarity*).
2. $Sim_{\psi_p^{s_0}, s_0, n}$ is the similarity obtained using the timed strong similarity s_0 and the lcp-selector function $\psi_p^{s_0}$ (*lcp-similarity*). \square

The following result states that for those MLTAPNs that are timed bisimilar we have $Sim_{\psi_p^{s_0}, s_0, n}$ equal to 1. Let us first introduce timed bisimilarity for MLTAPNs.

Definition 13. (Timed bisimilarity)

Let $(N_1, M_{0,1}), (N_2, M_{0,2})$ be two MLTAPNs. We say that they are *timed bisimilar* if and only if there is a relation $\rho \subseteq [(N_1, M_{0,1}) \times [(N_2, M_{0,2})]]$ such that:

1. $(M_{0,1}, M_{0,2}) \in \rho$.
2. $\forall (M_1, M_2) \in \rho$:
 - (a) $\forall t_1 \in T_1$, such that $M_1[t_1]_0 M'_1 \Rightarrow \exists t_2 \in T_2, M_2[t_2]_0 M'_2$, with $(M'_1, M'_2) \in \rho$ and $\lambda(t_1) = \lambda(t_2)$.
 - (b) $\forall t_2 \in T_2$, such that $M_2[t_2]_0 M'_2 \Rightarrow \exists t_1 \in T_1, M_1[t_1]_0 M'_1$, with $(M'_1, M'_2) \in \rho$ and $\lambda(t_1) = \lambda(t_2)$.
3. $\forall (M_1, M_2) \in \rho, \forall x \in \mathbb{N}^+ : (age(M_1, x), age(M_2, x)) \in \rho$. \square

Theorem 3. Let $(N_1, M_{0,1}), (N_2, M_{0,2})$ be two bounded timed bisimilar MLTAPNs and $n \in \mathbb{N}^+$. Then, we have $Sim_{\psi_p^{s_0}, s_0, n}((N_1, M_{0,1}), (N_2, M_{0,2})) = 1$.

Proof: Immediate, because for every $\sigma_1 \in BTTS_n(N_1, M_{0,1})$ we know that every movement in σ_1 must have its counterpart in $(N_2, M_{0,2})$ since they are timed bisimilar. Thus, we have this same timed occurrence sequence in $(N_2, M_{0,2})$, and conversely for the sequences $\sigma_2 \in BTTS_n(N_2, M_{0,2})$. \square

However, the converse is not true, we can have $Sim_{\psi_p^{s_0}, s_0, n}((N_1, M_{0,1}), (N_2, M_{0,2})) = 1$ for all $n \in \mathbb{N}^+$, whereas $(N_1, M_{0,1})$ and $(N_2, M_{0,2})$ are not timed bisimilar. Figure 7 shows two MLTAPNs for which $Sim_{\psi_p^{s_0}, s_0, n}((N_1, M_{0,1}), (N_2, M_{0,2})) = 1$ ($\forall n \geq 1$) that are not timed bisimilar.

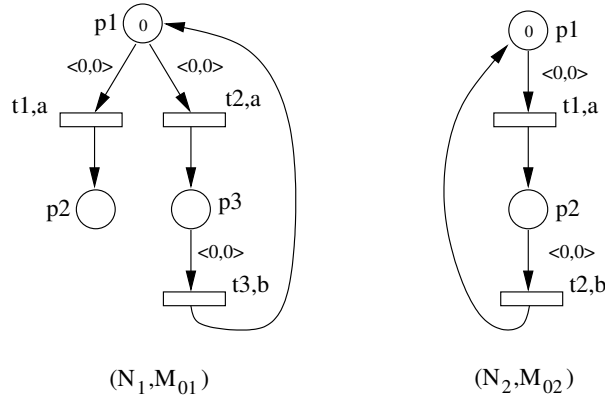


Figure 7: MLTAPNs that are not timed bisimilar.

We can see from this example that *lcp*-similarity fails to distinguish these MLTAPNs due to the conflict with both transitions labeled with the same action *a*. This occurs because similarities are based on timed traces, and every timed trace of $(N_1, M_{0,1})$ has its counterpart in $(N_2, M_{0,2})$, and conversely. It can be a matter of further research to define a similarity that can also compare the terminal timed traces of both MLTAPNs, where terminal timed traces are those for which only time can elapse after a final action. With this modification we could distinguish the MLTAPNs in Figure 7.

6. Evaluation

As mentioned above, the similarity based on the *lcp*-selector in general provides better results than that based on the *btt*s-selector function, because it allows the other MLTAPN to obtain timed occurrence sequences not belonging to its $BTTS_n$ that can further simulate the behavior of the provided sequences. This conclusion is now validated with some examples. With these examples we also show how the similarities capture the different behavior of MLTAPNs

that apparently look the same. These empirical results have been obtained by a prototype that is part of the TPAL framework [29].

Table 1 shows the results obtained for both similarities in five examples, taking $n = 1, 2, 3, 4$ (first column). The MLTAPNs that have been compared, $\{(N_{i,1}, N_{i,2})\}_{i=1}^5$, are those depicted in Figure 8. For each value of n we show the $BTTS_n$ size for both MLTAPNs, the *btt-similarity* and the *lcp-similarity*.

In the first example, we have considered two MLTAPNs that have the same behavior, (N_{11}, N_{12}) . They are actually timed bisimilar, and they can perform an arbitrary number of actions a in time 0 (even infinite), but if a time unit elapses no more actions will be performed and only time elapsing will be possible. Note that *lcp-similarity* is 1 for all values of n , as Theorem 3 stated. In contrast, *btt-similarity* fails to identify this same behavior, although it provides higher and higher values as n increases.

In the second example, (N_{21}, N_{22}) , we are considering a mutual exclusion problem with two processes that eventually request a shared resource, but process 1 must wait for at least one time unit before starting to request the resource, while process 2 can request the resource immediately. Both processes use the resource for exactly one time unit and then release the resource. At first sight the two MLTAPNs seem to have the same behavior, but *lcp-similarity* provides lower and lower values as n increases (0.35 for $n = 4$) and *btt-similarity* also gives values near 0.30. A thorough inspection of their behavior reveals the difference, namely that N_{21} has two different places for the activation of the corresponding request transition, which can have different token ages, in contrast to the only shared precondition place in the case of N_{22} . Thus, after process 2 has released the resource in N_{22} , process 1 must wait for at least one time unit to obtain the resource, whereas in N_{21} process 1 could immediately use the resource, because the token in $p1$ would be already old enough to allow the firing of $t1$.

Notice that (N_{31}, N_{32}) only differ from the previous ones in the time intervals labeling the arcs from $p1$ to $t1$ and $p1$ to $t3$ in N_{31} , and from $p1$ to $t1$ and $t3$ in N_{32} . All of these arcs are now labeled with the interval $[0, 0]$. This slight modification is enough to make both MLTAPNs timed bisimilar, so *lcp-similarity* is 1 for all n . However, *btt-similarity* once again fails to identify the two MLTAPNs, providing values close to 0.45.

In the case of (N_{41}, N_{42}) we have two MLTAPNs that clearly have a different behavior, because N_{41} must perform sequences of actions $a; b$ with a possible delay between them of one time unit, or in the event of a greater delay without performing the corresponding action the only possible behavior would be time elapsing. However, N_{42} can perform either action a or action b as long as these executions are immediate or separated by one time unit. In the event of a greater delay, the token in p becomes dead, and no further action can be performed, only time elapsing would be possible. In this case, *btt-similarity* gives values close to 0.50, while *lcp-similarity* starts with 1.0 for $n = 1$ and decreases to 0.85 for $n = 4$. This apparently high value is a consequence of considering not only the actions, but also the delays between them. In addition, N_{42} can obviously perform every sequence of N_{41} .

In the final case we have considered N_{41} again, but introducing a new transi-

		(N_{11}, N_{12})	(N_{21}, N_{22})	(N_{31}, N_{32})	(N_{41}, N_{42})	(N_{41}, N_{51})
1	$ BTT S_1 _1$	1	56	26	3	3
	$ BTT S_2 _1$	2	5	4	1	7
	<i>btt</i> -sim	0.66	0.32	0.44	0.50	0.67
	<i>lcp</i> -sim	1.0	0.80	1.0	1.0	0.84
2	$ BTT S_1 _2$	2	2786	490	15	15
	$ BTT S_2 _2$	4	44	30	5	35
	<i>btt</i> -sim	0.78	0.30	0.45	0.53	0.78
	<i>lcp</i> -sim	1.0	0.57	1.0	0.92	0.90
3	$ BTT S_1 _3$	3	127088	9972	63	63
	$ BTT S_2 _3$	6	371	208	21	147
	<i>btt</i> -sim	0.82	0.28	0.45	0.50	0.85
	<i>lcp</i> -sim	1.0	0.44	1.0	0.85	0.93
4	$ BTT S_1 _4$	4	5749802	214278	255	255
	$ BTT S_2 _4$	8	3224	1470	85	595
	<i>btt</i> -sim	0.84	0.26	0.46	0.47	0.89
	<i>lcp</i> -sim	1.0	0.35	1.0	0.85	0.95

Table 1: Experimental results.

tion $t3$ that can fire after either one or two time units from the firing of $t1$. Thus, we have a rather similar, but obviously different behavior for both MLTAPNs. Actually, both similarities tend to provide results around 0.9 in this case.

7. Conclusions, Discussion and Future Work

In this paper we have defined a reduced version of the timed reachability graph for bounded MLTAPNs, and we have introduced a general concept of similarity, on the basis of both a selector function and a similarity function for timed occurrence sequences. We have then introduced two selector functions (*btt*s and *lcp*) and a strong similarity function for timed occurrence sequences, and using them we have defined two strong timed similarities. We have concluded that *lcp*-similarity provides in general better results, because the other MLTAPN can provide better sequences to simulate the behavior of the given ones. The main problem is the state explosion problem, which becomes even greater when introducing large time intervals in the models, because we have many possible non-deterministic behaviors for each state of the system. Thus, even for rather simple examples, the computation of $|BTT S|_n$ becomes intractable and so also the similarity computation. Therefore, in this paper we have only presented the theoretical foundations of timed similarities, so further research must be carried out in order to have at least some approximate results for intractable systems.

Further research can also focus on weak timed similarities, in which time is not considered as relevant as the actions performed, so the similarity can weigh

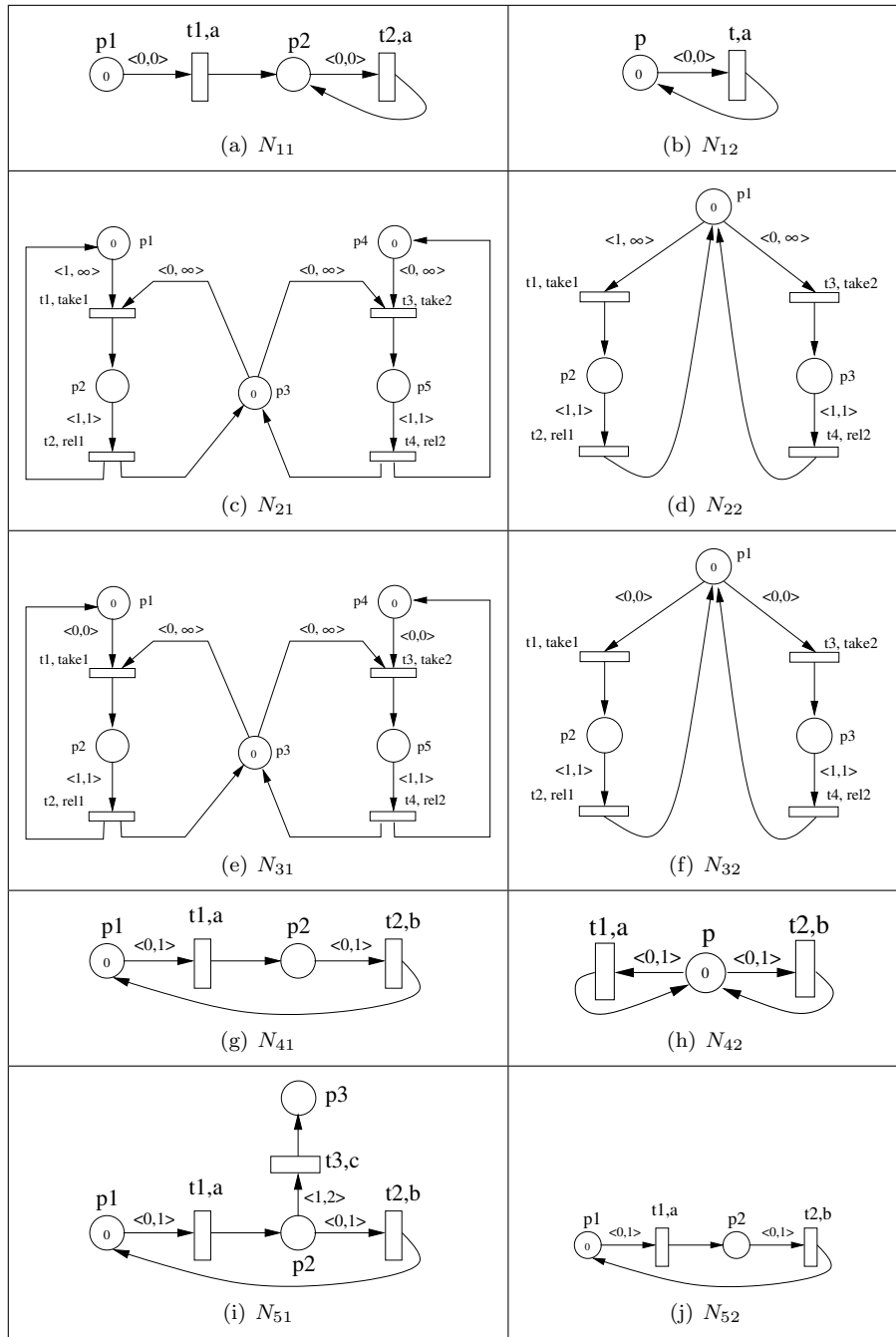


Figure 8: MLTAPNs for evaluation.

the impact of time discrepancies in some way, instead of stopping at the first time mismatching.

References

- [1] W.M.P. van der Aalst. *Interval Timed Coloured Petri Nets and their Analysis*. Lecture Notes in Computer Science, vol. 691, pp. 451-472. 1993.
- [2] P.A. Abdulla and A. Nylén. *Timed Petri Nets and BQOs*. Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 2075, pp. 53-70. 2001.
- [3] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte and A. Cumani. *On Petri Nets with Stochastic Timing*. Proc. of the International Workshop on Timed Petri Nets, IEEE Computer Society Press, pp. 80-87. 1985.
- [4] A.K. Alves de Medeiros, W.M.P. van der Aalst, A.J.M.M. Weijters. *Quantifying Process Equivalence Based on Observed Behavior*. Data & Knowledge Engineering journal, vol. 64(1), pp. 55-74. 2008.
- [5] M. Andersen, H.G. Larsen, J. Srba, M.G. Sørensen, J.H. Taankvist. *Verification of Liveness Properties on Closed Timed-Arc Petri Nets*. In 8th Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS'12), Lecture Notes in Computer Science, vol. 7721, pp. 69-81. 2013.
- [6] J. Bae, L. Liu, J. Caverlee, L. Zhang, H. Bae. *Development of Distance Measures for Process Mining, Discovery and Integration*. International Journal of Web Service Research, vol. 4(4), pp. 1-17. 2007.
- [7] S.V. Birch, T.S. Jacobsen, J.J. Jensen, C. Moesgaard, N.N. Samuelson, J. Srba. *Interval Abstraction Refinement for Model Checking of Timed-Arc Petri Nets*. In 12th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'14), Lecture Notes in Computer Science, vol. 8711, pp. 237-251, 2014.
- [8] T. Bolognesi, F. Lucidi and S. Trigila. *From Timed Petri Nets to Timed LOTOS*. Proc. of the 10th International IFIP WG6.1 Symposium on Protocol Specification, Testing and Verification, North-Holland, pp. 1-14. 1990.
- [9] F.D.J. Bowden. *Modelling Time in Petri Nets*. Proc. Second Australia-Japan Workshop on Stochastic Models. 1996.
<http://www.itr.unisa.edu.au/fbowden/pprs/stomod96/>.
- [10] A. Cerone and A. Maggiolo-Schettini. *Time-based Expressivity of Time Petri Nets for System Specification*. Theoretical Computer Science (216)1-2, pp. 1-53. 1999.

- [11] L. Chavarría-Baéz, X. Li. *A Petri Net-Based Metric for Active Rule Validation*. In 23rd IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society Press, pp. 922–923. 2011.
- [12] A. David, L. Jacobsen, M. Jacobsen, K.Y. Jørgensen, M.H. Møller, J. Srba. *TAPAAL 2.0: Integrated Development Environment for Timed-Arc Petri Nets*. 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'12), Lecture Notes in Computer Science, vol. 7214, pp. 492–497. 2012.
- [13] D. de-Frutos, V. Valero and O. Marroquín. *Decidability of Properties of Timed-Arc Petri Nets*. Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 1825, pp. 187–206. 2000.
- [14] R.M. Dijkman, M. Dumas, L. García-Bañuelos. *Graph Matching Algorithms for Business Process Model Similarity Search*. 7th International Conference on Business Process Management (BPM), Lecture Notes in Computer Science, vol. 5701, pp. 48–63. 2009.
- [15] F. Emmert-Streib, M. Dehmer, Y. Shi. *Fifty Years of Graph Matching, Network Alignment and Network Comparison*. Information Sciences, Science Direct, vol. 346–347, pp. 180–197. 2016.
- [16] H.-M. Hanisch. *Analysis of Place/Transition Nets with Timed-Arcs and its Application to Batch Process Control*. Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 691, pp. 282–299. 1993.
- [17] L. Jacobsen, M. Jacobsen, M.H. Møller, J. Srba. *Verification of Timed-Arc Petri Nets*. In SOFSEM 2011, Theory and Practice of Computer Science, Lecture Notes in Computer Science, vol. 6543, pp. 46–72. 2011.
- [18] J.B. Kenneth, J. Srba. *An Efficient Translation of Timed-Arc Petri Nets to Networks of Timed Automata*. Proc. of Formal Methods and Software Engineering, Lecture Notes in Computer Science, vol. 5885, pp. 698–716. 2009.
- [19] M. Kunze, M. Weidlich, M. Weske. *m³ - A Behavioral Similarity Metric for Business Processes*. In 3rd Central-European Workshop on Services and Their Composition (ZEUS'11), pp. 89–95. 2011.
- [20] C. Mao. *Complexity Analysis for Petri Net-Based Business Process in Web Service Composition*. In 5th International Symposium on Service Oriented System Engineering, IEEE Computer Society, pp. 193–196. 2010.
- [21] P. Merlin. *A Study of the Recoverability of Communication Protocols*. PhD. Thesis, Univ. of California. 1974.
- [22] M. Nielsen, V. Sassone and J. Srba. *Properties of Distributed Timed-Arc Petri Nets*. Proc. of the 21st International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'01), Lecture Notes in Computer Science, vol. 2245, pp. 280–291. 2001.

- [23] C. Ramchandani. *Performance Evaluation of Asynchronous Concurrent Systems by Timed Petri Nets*. PhD. Thesis, Massachusetts Institute of Technology, Cambridge. 1973.
- [24] M. Rosemann. *Potential Pitfalls of Process Modelling: Part B*. Business Process Management Journal, vol. 12(3), pp. 377-384. 2006.
- [25] J. Srba. *Timed-Arc Petri Nets vs. Networks of Timed-Automata*. Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 3536, pp. 385-402. 2005.
- [26] V. Valero, D. de Frutos Escrig, and F. Cuartero. *On Non-Decidability of Reachability for Timed-Arc Petri Nets*. Proc. 8th. International Workshop on Petri Nets and Performance Models (PNPM'99), pp. 188-196. 1999.
- [27] V. Valero and H. Macià. *Removing Dead Transitions in Timed-Arc Petri Nets*. In Mathematical and Computer Modelling of Dynamical Systems, vol. 15(1). pp. 69–82. 2009.
- [28] V. Valero, J.J. Pardo and F. Cuartero. *Translating TPAL Specifications into Timed-Arc Petri Nets*. Application and Theory of Petri Nets, Lecture Notes in Computer Science, vol. 2360, pp. 414-433. 2002.
- [29] V. Valero, J.J. Pardo. *TPAL tool*. Computer Science Department, University of Castilla-La Mancha.
<http://www.dsi.uclm.es/investigacion/retics/tools/tools.htm>
- [30] J. Wang, T. He, L. Wen, N. Wu, A.H.M. ter Hofstede, J. Su. *A Behavioral Similarity Measure Between Labeled Petri Nets Based on Principal Transition Sequences*. On the Move to Meaningful Internet Systems (OTM'10), Lecture Notes in Computer Science, vol. 6426, pp. 394–401. 2010.
- [31] Z. Wang, L. Wen, J. Wang, S. Wang. *TAGER: Transition-Labeled Graph Edit Distance Similarity Measure on Process Models*. On the Move to Meaningful Internet Systems, OTM 2014 Conferences. Lecture Notes in Computer Science, vol. 8841, pp. 184–201. 2014.
- [32] H. Zha, J. Wang, L. Wen, C. Wang, J. Sun. *A Workflow Net Similarity Measure Based on Transition Adjacency Relations*. In Computers and Industry, vol. 61, pp. 463–471. 2010.