



Proceedings of the
11th International Workshop on Graph Transformation and
Visual Modeling Techniques
(GTVMT 2012)

Algebraic Approach to Timed Petri Nets

Karsten Gabriel, Pascal Lingnau, Claudia Ermel

14 pages

Algebraic Approach to Timed Petri Nets

Karsten Gabriel*, Pascal Lingnau, Claudia Ermel

Technische Universität Berlin, Germany

kgabriel@cs.tu-berlin.de, pasling@cs.tu-berlin.de, claudia.ermel@tu-berlin.de

Abstract: One aspect often needed when modelling systems of any kind is time-based analysis, especially for real-time or in general time-critical systems. Algebraic place/transition (P/T) nets do not inherently provide a way to model the passing of time or to restrict the firing behaviour with regards to passing time. In this paper, we present an extension of algebraic P/T nets by adding time durations to transitions and timestamps to tokens. We define categories for different timed net classes and functorial relations between them. Our first result is the definition of morphisms preserving firing behaviour for all timed net classes. As second result, we define structuring techniques for timed P/T nets in a way that our category fulfills the properties of \mathcal{M} -adhesive systems, a general categorical framework for structuring and transforming high-level algebraic structures. We demonstrate our approach by applying it to model a real-time communication network.

Keywords: algebraic Petri nets, Petri nets with time, timed P/T nets, Petri net structuring technique, P/T net transformation

1 Introduction

Petri nets are a formalism widely used for modelling and analyzing systems and processes. First introduced by Carl Adam Petri in [Pet62], the notion of (*classical*) Petri nets (and place/transition (P/T) nets in particular) has been refined and extended over the time [Rei85, Rei10], also considering coloured tokens giving rise to the notion of high-level nets, like e. g. *Coloured Petri nets* [Jen92, JKW07]. Since Petri nets do not inherently provide a way to model the passing of time or to restrict the firing behaviour with regards to passing time, a number of approaches have been established that enhance classical Petri nets by notions of time, such as *time Petri nets* [BD91], *deterministic timed Petri nets* [BH07] and *timed coloured Petri nets* [JK09].

In the nineties, a main objection against Petri nets as specification technique was that Petri nets lack abstraction, data-type handling, refinement and structuring. Hence, composition and abstraction mechanisms for Petri nets have been formalized in an algebraic setting [MM90, ER97], viewing an *algebraic P/T net* as a directed graph whose set of nodes is the free commutative monoid P^\oplus generated by the set P of places, and transitions are the arcs between them. Note that there is a one-to-one correspondence of classical and algebraic P/T nets. With the definition of morphisms between algebraic Petri nets categories of Petri nets can be defined with and without initial markings. The results provide a formal basis to reason on abstraction and refinement, on

* This author is sponsored by the Integrated Graduate Program on Human-Centric Communication at Technische Universität Berlin

integration and decomposition of nets, and on semantics given by net processes, at an abstract level using methods from category theory [EM96, EPR94]. Since then, new algebraic Petri net categories, as well as extensions of existing Petri net categories have been developed, including P/T nets with individual tokens (PTI nets) [MGH11], and Algebraic High-Level (AHL) nets [EHP⁺02, Ehr04].

Unfortunately, in contrast to classical Petri nets, up to now there exist no extensions of algebraic Petri nets by concepts for modelling the passing of time. Adding such a concept to the formalism would allow us e. g. to establish structuring techniques for Petri nets with time in a categorical setting, and to re-use existing results for algebraic Petri nets with respect to rule-based Petri net reconfiguration [EG11, PEHP08].

In this paper, we provide a definition of algebraic, timed P/T nets, which enables us to define categories for different classes of timed nets, allowing the definition of the structuring techniques *gluing* and *restriction*, and morphisms to specify processes for timed P/T nets. Moreover, timed P/T nets can be reconfigured using timed P/T net transformation rules in the well-known algebraic double pushout (DPO) approach [EEPT06]. In our formalization, tokens carry timestamps, usually denoting the earliest point in time when this token may be consumed by a transition. A global clock models the current time and influences the activation of transitions. Tokens generated in a firing step get a timestamp computed from the current firing time (the global clock) and the time duration of the transition.

The paper is structured as follows: After considering existing approaches that extend Petri nets by time in Section 2, we present our running example, a real-time communication network, in Section 3. In Section 4, we formalize our algebraic notion of timed P/T nets. Categories of timed P/T net classes and their functorial relations are presented in Section 5, while structuring techniques and transformation are introduced in Section 6. We conclude with an outlook to future work in Section 7.

2 Extending Petri Nets by Time Modelling

There are many approaches for extending Petri nets by time. Our approach is based on the approach of timed Coloured Petri nets (timed CPNs) [JK09], where the well-known high-level Petri net variant CPNs [Jen92] has been extended by assigning a *duration* to a transition and so-called *timestamps* to each token, indicating the earliest point in time when a token can be used by a transition. A transition that fires adds the duration time of the transition to each created token's timestamp, so that in general, these tokens cannot be used immediately, but only after the time the transition takes has passed. The timed CPN firing behaviour requires that a transition fires at the earliest point in time at which it is activated. During simulation, the global clock is monotonically increasing, as there is no possibility of a transition being activated at a point in time that has already passed. In our approach, we do not have this limitation of the firing behaviour, so in principle, simulations become possible that explore the past (for more details we refer to [LGM12]). The tool CPN TOOLS provides means to design and analyse timed CPNs, including state-space analysis and model checking techniques [JKW07].

In Time Petri nets (TPNs) [BD91], two labels are assigned to each transition, denoting the time that has to pass before the transition can fire after being enabled (earliest firing time), and

the maximal time it may be enabled until it has to fire (latest firing time). This firing behaviour is significantly different from that of timed CPNs, allowing for more control over the model behaviour. TPNs can be analysed using a state-space approach [GLMR05].

Deterministic timed Petri nets [BH07] pursue a rather unique approach for firing behaviour by introducing a delay between removal and creation of tokens. In addition, each place has a designated delay, denoting the time before a created token can be consumed from that place.

In time environment-relationship (TER) nets [GMMP91], time is considered as token attribute (*chronos*) with special behaviour. Token timestamps are set by specific action relations associated to transitions, which also require time monotonicity of firing sequences. This approach has inspired the definition of graph transformation systems with time [GVH04], since considering time as attribute one may re-use the formal framework of attributed graph transformation.

None of the approaches listed above are based on the general algebraic framework that allows for the definition of structuring and reconfiguration techniques for P/T nets with time.

3 Running Example: A Communication Network Infrastructure

We consider a communication network of four routers connected in a loop of forwarding transitions (see Figure 1a). Each client is connected to a router via *send* and *receive* transitions that model sending and receiving of packets from/to the router. A router allows only a certain number of open connections at a time. This is modelled by the number of tokens on the corresponding *ready* places which represent the maximum number of concurrent connections the router is able to maintain. Hence, a router is ready to forward a packet to the next router only if there are enough tokens on the corresponding *ready* places. Obviously, in the untimed P/T net in Figure 1a, the *ready* tokens are consumed and immediately produced again whenever a forward action occurs. Due to the instant production of the tokens upon firing of a *forward* transition, the number of tokens on the *ready* places has no effect in an untimed P/T net, even if many forward transactions occur at the same time. Therefore, we need a way to express the duration such a transaction takes, which is possible using timed P/T nets. The assigned firing durations restrict the firing sequences possible in the simulation of the net, which yields a behaviour that is more faithful to that of a real-time communication network.

In the timed P/T net variant, shown in Figure 1b, each token of a marking carries a timestamp (by which it is represented in the illustration), denoting the earliest point in time when this token can be consumed by a transition. A global clock is employed in order to be able to specify “when” transitions are activated and when they fire, with timestamps and clock values being real numbers. The arc inscriptions represent the time duration of the action associated with the connected transition. Note that the arc inscriptions are sums of time values in general, so an output arc creating two tokens would be a sum (\oplus) of two time values.¹ Analogously to untimed P/T nets, an arc with no inscription means that there is a single token created or consumed with no time delay, i. e. an arc without inscription is equivalent to an arc with a single zero. When firing a transition at a global clock value (not shown in Figure 1b), the newly created tokens get assigned a timestamp that is the sum of that clock value and the respective time duration specified by the output arc. For example, transition *send2* creates a token with a timestamp of 300 time

¹ Note that in our example, we have only sums of size one as arc inscriptions.

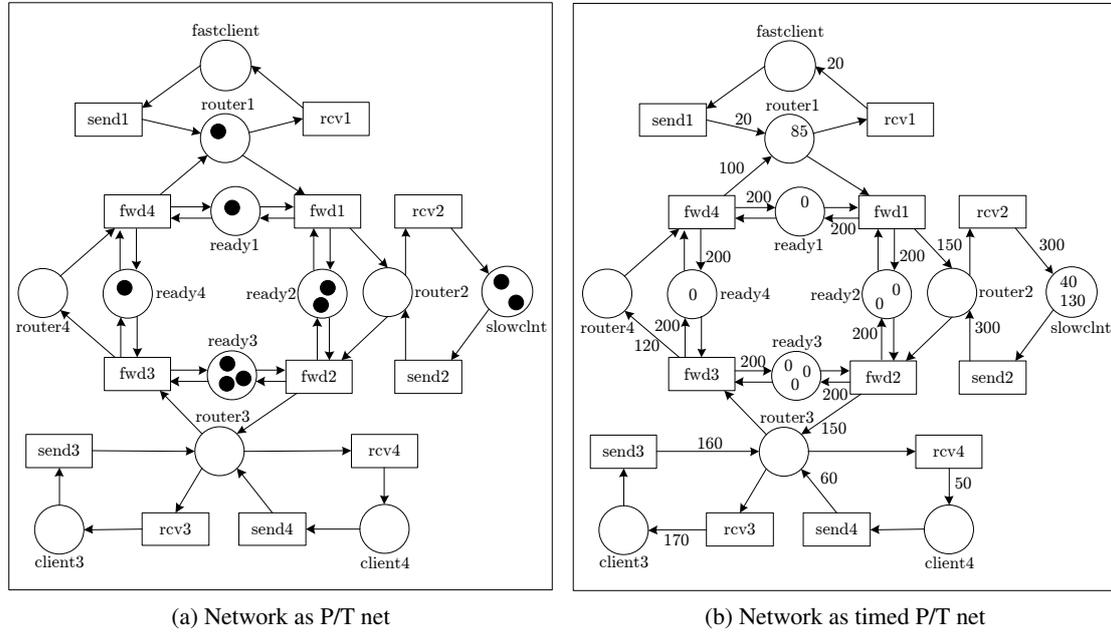


Figure 1: Network as P/T net and timed P/T net

units later than the global clock value at the time of firing.

As further example, consider the firing of transition $fwd1$ at clock value 85, the earliest clock value at which the token on $router1$ can be used according to its timestamp. The firing removes that token and creates a token on $router2$ with a timestamp of 235 ($= 85 + 150$). Also, the tokens with timestamp zero on $ready1$ and $ready2$ are removed and replaced by tokens with timestamp 285. Therefore, transition $fwd2$ is enabled not earlier than at clock value 285, since this is the timestamp of the token on $ready2$, regardless of the timestamp of 235 of the token on $router2$. In contrast, in the untimed P/T net in Figure 1a, transition $fwd2$ could fire immediately, as the token on $ready2$ simply is recreated when $fwd1$ is fired.²

4 Algebraic Formalisation of Timed P/T Nets

We define timed P/T nets and systems based on algebraic P/T nets, adding time values to the markings (including pre and post domains). Timed P/T states are an extension of timed P/T systems, incorporating a global clock value.

In the following X^\oplus denotes the free commutative monoid over the set X . Note that $S \in X^\oplus$ is a formal sum $S = \sum_{i=1}^n \lambda_i x_i$ with $\lambda_i \in \mathbb{N}$ and $x_i \in X$ meaning that we have λ_i copies of x_i in S , and for $S' = \sum_{i=1}^n \lambda'_i x_i$ we have $S \oplus S' = \sum_{i=1}^n (\lambda_i + \lambda'_i) x_i$. Moreover, a sum $S_0 \in X^\oplus$ is a subsum of S , written $S_0 \leq S$ if there exists a sum $S' \in X^\oplus$ such that $S = S_0 \oplus S'$.

² For further examples where pre-arcs are inscribed by time values, see [LGM12].

Definition 1 (Timed P/T Net Classes and Timed Marking)

1. A *timed P/T net* or TPT net $TN = (P, T, pre, post)$ consists of a set P of places, a set T of transitions and functions $pre, post : T \rightarrow (P \times \mathbb{R})^\oplus$.
2. A *timed P/T system* or TPT system is a tuple (TN, M) with timed P/T net TN and a marking M of TN which is an element $M \in (P \times \mathbb{R})^\oplus$.
3. A *timed state* or TPT state is a 3-tuple (TN, M, τ) with timed P/T net TN , a marking M of TN and a global clock value $\tau \in \mathbb{R}$.

Remark 1 (Relation to Untimed P/T Nets) *Note that the definition of timed P/T nets is quite similar to the definition of untimed algebraic nets [ER97]. The only difference is that in the timed case the pre and post arcs are equipped with time-values. Accordingly, in a timed marking each token consists of a time-value, called timestamp.*

Example 1 (Timed P/T System) *Figure 1b shows a timed P/T net $TN = (P, T, pre, post)$ with places P and transitions T as described in Section 3. The pre- and post-domains of fwd1, for example, are: $pre(\text{fwd1}) = (\text{ready1}, 0) \oplus (\text{ready2}, 0) \oplus (\text{router1}, 0)$, $post(\text{fwd1}) = (\text{ready1}, 200) \oplus (\text{ready2}, 200) \oplus (\text{router2}, 150)$.*

The depicted marking is $M = (\text{router1}, 85) \oplus (\text{slowCnt}, 40) \oplus (\text{slowCnt}, 130) \oplus (\text{ready1}, 0) \oplus 2 \cdot (\text{ready2}, 0) \oplus 3 \cdot (\text{ready3}, 0) \oplus (\text{ready4}, 0)$.

Next, we define *time delays* as relations between markings, to decide whether a transition can be fired with respect to the timestamps of the tokens involved.

Definition 2 (Time-Delay) Let $M_1, M_2 \in (P \times \mathbb{R})^\oplus$ be two timed markings of TPT net $TN = (P, T, pre, post)$. We define the following two types of delays:

- M_1 is a *location-strict delay* of M_2 , written $M_1 \stackrel{\leftarrow}{\leq} M_2$, if $\pi_p^\oplus(M_1) = \pi_p^\oplus(M_2)$ and for all $p \in P$: $M_1[p] \geq M_2[p]$, with $M[p]$ being the list of timestamps of all tokens on place p in ascending sorted order, and $\pi_p^\oplus(M)$ is the projection that “forgets” the time-values, called *location* of M . We call the timestamps in M_1 *later* than those in M_2 .
- M_1 is a *delay* of M_2 , written $M_1 \stackrel{\leftarrow}{\leq} M_2$, if there exists a subsum $M'_2 \leq M_2$, so that $M_1 \stackrel{\leftarrow}{\leq} M'_2$.³

Example 2 (Time-Delay) *Consider marking $M_2 = (\text{slowCnt}, 40) \oplus (\text{slowCnt}, 130)$ of the TPT net shown in Figure 1b. Then, $M_1 = (\text{slowCnt}, 50) \oplus (\text{slowCnt}, 150)$ is a location-strict delay of M_2 , i. e. $M_1 \stackrel{\leftarrow}{\leq} M_2$. For the marking $M_3 = (\text{slowCnt}, 40) \oplus (\text{slowCnt}, 130) \oplus (\text{router1}, 85)$ we have that M_1 is a delay of M_3 , i. e. $M_1 \stackrel{\leftarrow}{\leq} M_3$, since there is $M_2 \leq M_3$ with $M_1 \stackrel{\leftarrow}{\leq} M_2$.*

Now we can define the firing behaviour of timed P/T nets. A transition is enabled under a marking $S \leq M$ of a timed state (where S is called *selection*), if the global clock is late enough in order to satisfy the timestamps of the tokens in the selection. The follower marking is then

³ The symbols chosen for delays are constructed as follows: The bottom comparator indicates which marking is larger (or that they are of equal location), while the arrow above points to the marking with the later timestamps.

calculated like for algebraic P/T nets, but now includes the calculation of the newly created tokens' timestamps by adding the time duration of the transition to the current global clock value.

Definition 3 (Firing Behaviour) Given a timed P/T state (TN, M, τ) with timed P/T net $TN = (P, T, pre, post)$, marking M of TN , global clock value τ , and a selection $S \leq M$. Then $t \in T$ is *enabled under* (S, τ) if $pre(t)^{+\tau}$ is a location-strict delay of S , i. e. $pre(t)^{+\tau} \stackrel{\leftarrow}{=} S$, where $pre(t)^{+\tau}$ means the addition of τ to every timestamp in the marking $pre(t)$.

For $t \in T$ enabled under (S, τ) , there is a *firing step* $(TN, M, \tau) \xrightarrow{(t, S)} (TN, M', \tau)$, where the follower marking M' is given by $M' = M \ominus S \oplus post(t)^{+\tau}$.

Like in algebraic P/T nets, we can link successive firing steps in a firing sequence. Since a firing step does not advance the global clock, so-called *timesteps* of the form $(TN, M, \tau) \xrightarrow{\Delta\tau} (TN, M, \tau + \Delta\tau)$ are inserted between two firing steps where the global clock is changed by $\Delta\tau \in \mathbb{R}$. The differentiation between timesteps and firing steps allows us to model actions that overlap in time.

Definition 4 (Firing Sequence) Given a timed P/T state (TN, M_0, τ_0) with timed P/T net $TN = (P, T, pre, post)$, marking M_0 of TN , global clock value τ_0 , and $t_i \in T$ enabled under (S_i, τ_i) for $i \in \{0, \dots, n-1\}$ and $S_i \leq M_i$. Then,

$$Seq = (TN, M_0, \tau_0) \xrightarrow{(t_0, S_0)} (TN, M_1, \tau_0) \xrightarrow{\Delta\tau_0} (TN, M_1, \tau_1) \\ \xrightarrow{(t_1, S_1)} \dots (TN, M_{n-1}, \tau_{n-2}) \xrightarrow{\Delta\tau_{n-2}} (TN, M_{n-1}, \tau_{n-1}) \xrightarrow{(t_{n-1}, S_{n-1})} (TN, M_n, \tau_{n-1})$$

is a *firing sequence* in net TN , if for all $i \in \{0, \dots, n-1\}$: $M_i \xrightarrow{(t_i, S_i, \tau_i)} M_{i+1}$ is a firing step.

The following is a shorter variant of the same firing sequence, omitting the timesteps, with the clock value at the time of firing included in the firing step notation:

$$Seq = M_0 \xrightarrow{(t_0, S_0, \tau_0)} M_1 \xrightarrow{(t_1, S_1, \tau_1)} \dots \xrightarrow{(t_{n-1}, S_{n-1}, \tau_{n-1})} M_n.$$

Note that in general there is no constraint on the global clock values. We can, however, enforce different restrictions on firing sequences in order to achieve a certain behaviour:

- Seq is called *time-monotonic*, if for $0 \leq i < n$, there is $\tau_i \leq \tau_{i+1}$.
- Seq employs *eager firing*, if for all firing steps $M_i \xrightarrow{(t_i, S_i, \tau_i)} M_{i+1}$ in Seq , there is no firing step $M_i \xrightarrow{(t'_i, S'_i, \tau'_i)} M'_{i+1}$ with $\tau'_i < \tau_i$.⁴

Example 3 (Firing Behaviour) In the timed P/T net depicted in [Figure 1b](#), fwd1 is enabled at $\tau = 85$ under the selection $S = (ready1, 0) \oplus (ready2, 0) \oplus (router1, 85)$, since $pre(t)^{+\tau} \stackrel{\leftarrow}{=} S$, i. e. $pre(fwd1)^{+85} = (ready1, 0 + 85) \oplus (ready2, 0 + 85) \oplus (router1, 0 + 85) \stackrel{\leftarrow}{=} S$.

Therefore, fwd1 can fire at $\tau = 85$, leading to the follower marking

$$M' = M \ominus S \oplus post(t)^{+\tau} = (ready1, 285) \oplus (ready2, 0) \oplus (ready2, 285) \\ \oplus 3 \cdot (ready3, 0) \oplus (ready4, 0) \oplus (slowclnt, 40) \oplus (slowclnt, 130) \oplus (router2, 235).$$

⁴ To simulate timed CPNs, time-monotonic firing sequences with eager firing must be used.

5 Categories for Timed P/T Net Classes and Functorial Relations

In this section, we define the category of timed P/T nets. Timed P/T morphisms are defined in a way that they preserve firing behaviour, and thus in general the codomain net of a morphism is more liberal than the domain net. In the following, for a function $f_P : P_1 \rightarrow P_2$ between sets of places P_1 and P_2 , we use $f_{P \times \mathbb{R}}^\oplus$ to denote the extension $(f_P \times id_{\mathbb{R}})^\oplus : (P_1 \times \mathbb{R})^\oplus \rightarrow (P_2 \times \mathbb{R})^\oplus$ of f_P to markings over P_1 and P_2 .

Definition 5 (Category **TPTNets** of Timed P/T Nets) Given timed P/T-nets $TN_i = (P_i, T_i, pre_i, post_i)$, for $i \in \{1, 2\}$, a *timed P/T-net-morphism* (or short TPT morphism) $f : TN_1 \rightarrow TN_2$ is defined by $f = (f_P, f_T)$, with functions $f_P : P_1 \rightarrow P_2$ and $f_T : T_1 \rightarrow T_2$, such that for all $t \in T_1$:

$$pre_2 \circ f_T(t) \stackrel{\leftarrow}{=} f_{P \times \mathbb{R}}^\oplus \circ pre_1(t) \quad \text{and} \quad post_2 \circ f_T(t) \stackrel{\rightarrow}{=} f_{P \times \mathbb{R}}^\oplus \circ post_1(t).$$

A timed P/T-morphism f is called *time-strict* if for all $t \in T$ we have $pre_2 \circ f_T(t) = f_{P \times \mathbb{R}}^\oplus \circ pre_1(t)$ and $post_2 \circ f_T(t) = f_{P \times \mathbb{R}}^\oplus \circ post_1(t)$.

The category **TPTNets** of timed P/T nets consists of timed P/T nets as objects and timed P/T morphisms as morphisms. The composition and identities are defined componentwise for places and transitions in **Sets**.

Theorem 1 (Timed P/T Morphisms preserve Firing Behaviour) Given timed nets $TN_i = (P_i, T_i, pre_i, post_i)$ with $i = 1, 2$, with marking M of TN_1 , selection $S \leq M$ and a timed P/T morphism $f = (f_P, f_T), f : TN_1 \rightarrow TN_2$. Let $t \in T_1$ be enabled under S and $M \xrightarrow{(t, S, \tau)} M'$ a firing step in TN_1 with $M' = M \ominus S \oplus post_1(t)^{+\tau}$.

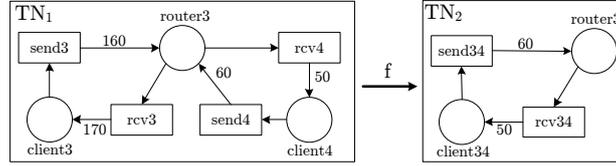
Then, there is a firing step $f_{P \times \mathbb{R}}^\oplus(M) \xrightarrow{(f_T(t), f_{P \times \mathbb{R}}^\oplus(S), \tau)} M''$ in TN_2 with $f_{P \times \mathbb{R}}^\oplus(M') \stackrel{\leftarrow}{=} M''$.

Proof Idea. Given a marking M and selection $S \leq M$ such that S is enabled under $t \in T_1$, then there is $pre(t)^{+\tau} \stackrel{\leftarrow}{=} S$. We also have a selection $f_{P \times \mathbb{R}}^\oplus(S) \leq f_{P \times \mathbb{R}}^\oplus(M)$, and since the order $\stackrel{\leftarrow}{=}$ is compatible with operations \oplus and \ominus as well as with the marking-extension $f_{P \times \mathbb{R}}^\oplus$ of f_P , we have $pre_2(f_T(t))^{+\tau} \stackrel{\leftarrow}{=} f_{P \times \mathbb{R}}^\oplus(pre_1(t))^{+\tau} = f_{P \times \mathbb{R}}^\oplus(pre_1(t)^{+\tau}) \stackrel{\leftarrow}{=} f_{P \times \mathbb{R}}^\oplus(S)$ which means that $f_T(t)$ is enabled under $(f_{P \times \mathbb{R}}^\oplus(S), \tau)$. Moreover, we obtain for the follower markings that $f_{P \times \mathbb{R}}^\oplus(M') = f_{P \times \mathbb{R}}^\oplus(M \ominus S \oplus post_1^{+\tau}(t)) = f_{P \times \mathbb{R}}^\oplus(M) \ominus f_{P \times \mathbb{R}}^\oplus(S) \oplus f_{P \times \mathbb{R}}^\oplus(post_1^{+\tau}(t)) \stackrel{\leftarrow}{=} f_{P \times \mathbb{R}}^\oplus(M) \ominus f_{P \times \mathbb{R}}^\oplus(S) \oplus post_2^{+\tau}(f_T(t)) = M''$. A detailed proof can be found in [LGM12]. \square

Example 4 (Timed P/T morphism and Preservation of Firing Steps) *Figure 2* shows timed P/T nets TN_1, TN_2 with morphism $f : TN_1 \rightarrow TN_2$, which maps places and transitions as indicated by their names. The morphism condition is fulfilled for all transitions, since in TN_2 , the time values in the postdomains of the transitions are lower than their counterparts in TN_1 , i. e. $post_2 \circ f_T(t) \stackrel{\rightarrow}{=} f_{P \times \mathbb{R}}^\oplus \circ post_1(t)$, while the values in the predomains are the same (zero).

Consider markings $M_1 = (\text{router3}, 100)$ of TN_1 and $M_2 = f_{P \times \mathbb{R}}^\oplus(M_1) = (\text{router3}, 100)$ of TN_2 and firing steps in both nets at $\tau = 100$ with selections $S_{1,2} = (\text{router3}, 100)$ and rcv3 , respectively rcv34 as the transitions to fire. The resulting markings are $M'_1 = (\text{client3}, 270)$ for TN_1 and $M'_2 = (\text{client34}, 150)$ for TN_2 , and we have $f_{P \times \mathbb{R}}^\oplus(M'_1) = (\text{router3}, 270) \stackrel{\leftarrow}{=} (\text{router3}, 150) = M'_2$.

We define a functor *TSkel* that maps timed P/T nets onto algebraic P/T nets by forgetting the

Figure 2: Timed P/T nets with timed morphism f

time values on the pre- and post-arcs. This allows us to analyse the underlying algebraic P/T net structure using existing techniques for P/T nets. An example is shown in Figure 1, where the P/T net in Figure 1a is the result of the application of $TSkel$ to the timed P/T net shown in Figure 1b.

Definition 6 (Functor $TSkel$) The functor $TSkel$ is defined as $TSkel : \mathbf{TPTNets} \rightarrow \mathbf{PTNets}$ with $TSkel(P, T, pre, post) = (P, T, pre^*, post^*)$, where $pre^*(t) = \pi_p^\oplus(pre(t))$, $post^*(t) = \pi_p^\oplus(post(t))$ for all $t \in T$ for the objects of $\mathbf{TPTNets}$. For the morphisms, we define $TSkel(f_P, f_T) = (f_P, f_T)$.

Remark 2 (Timed P/T Systems and States, Preservation of Firing Behaviour)

1. Analogously to timed P/T net morphisms, timed P/T system morphisms and timed P/T state morphisms are defined as timed P/T morphisms between the corresponding timed P/T nets with additional requirements, taking into account the marking and, in the case of states, also the global clock value. The categories of timed P/T systems and states are denoted as \mathbf{TPTSys} and $\mathbf{TPTState}$, respectively. For more details we refer to [LGM12].
2. In [LGM12] we also define a functor $TSkelSys$ that maps timed P/T systems to P/T systems, forgetting the timestamps in the corresponding markings.
3. There are functors $Abs : \mathbf{TPTSys} \rightarrow \mathbf{TPTStates}$ with $Abs(TN, M) = (TN, M, 0)$, and $Rel : \mathbf{TPTStates} \rightarrow \mathbf{TPTSys}$ with $Rel(TN, M, \tau) = (TN, M^{-\tau})$, establishing an equivalence of the categories \mathbf{TPTSys} and $\mathbf{TPTStates}$ as shown in [LGM12]. This means that every timed state can be expressed as a timed system by subtracting the global clock value from all tokens' timestamps, and $Abs \circ Rel$ can be used to obtain a "normal form" of timed states.
4. The functors $TSkel$ and $TSkelSys$ preserve firing behaviour, as shown in [LGM12].

6 Structuring and Transformation of Timed P/T Nets

In this section we define gluing and restriction constructions as structuring techniques for timed P/T nets. We also present an example transformation rule.

Intuitively, the gluing of timed P/T nets TN_1 and TN_2 over an interface TN_0 is a timed P/T net that contains copies of TN_1 and TN_2 which are glued together at places or transitions specified by the interface.

Definition 7 (Gluing of Timed P/T Nets) Given timed P/T nets $TN_i = (P_i, T_i, pre_i, post_i)$ with $i = 0 \dots 2$, and timed P/T morphism $f : TN_0 \rightarrow TN_1$ and $g : TN_0 \rightarrow TN_2$ where f is time-strict

(see Definition 5) and injective. Then, the gluing $TN_3 = (P_3, T_3, pre_3, post_3)$ of TN_1 and TN_2 along TN_0 , written $TN_3 = TN_1 +_{TN_0} TN_2$, with morphisms $f' : TN_2 \rightarrow TN_3$ and $g' : TN_1 \rightarrow TN_3$ is constructed as follows:

- P_3, T_3 are constructed as pushouts (2) and (3) in **Sets** (see Figure 3b), and
- $pre_3(t) = \begin{cases} f'_{P \times \mathbb{R}}(pre_2(t^*)) & \text{if } \exists t^* \in T_2, f'_T(t^*) = t \\ g'_{P \times \mathbb{R}}(pre_1(t')) & \text{if } \nexists t^* \in T_2, f'_T(t^*) = t \wedge \exists t' \in T_1 : g'_T(t') = t, \end{cases}$
and $post_3(t)$ is defined analogously.

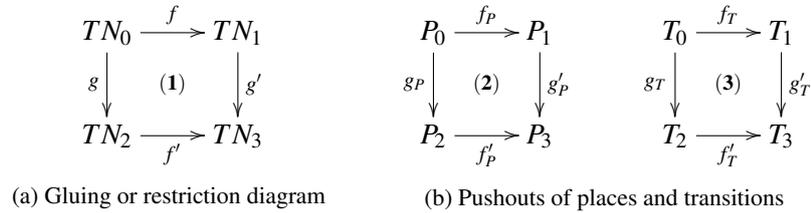


Figure 3: Gluing and restriction diagrams of timed P/T nets

Remark 3 (Pushout of Timed P/T Nets) In [LGM12] it is shown that the gluing construction defined above is also a pushout construction. This means that given the gluing (1) in Figure 3a of timed P/T nets, then (1) commutes and satisfies the following universal property: For all timed P/T nets TN_4 and timed P/T morphisms $h : TN_1 \rightarrow TN_4, k : TN_2 \rightarrow TN_4$ with $h \circ f = k \circ g$ there exists a unique timed P/T morphism $m : TN_3 \rightarrow TN_4$ such that $m \circ f' = k$ and $m \circ g' = h$.

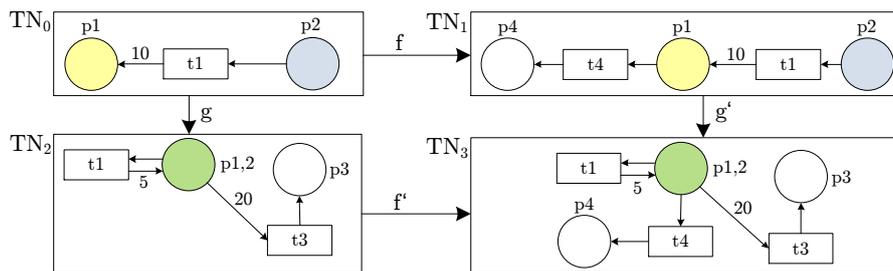


Figure 4: Gluing and restriction of timed P/T nets

Definition 8 (Restriction of Timed P/T Nets) Given a timed P/T morphism $g' : TN_1 \rightarrow TN_3$ and a subnet TN_2 of TN_3 with inclusion $f' : TN_2 \rightarrow TN_3$. The restriction $g : TN_0 \rightarrow TN_2$ of g' along f' is given by $g = g'|_{TN_0}$ where TN_0 is constructed as a subnet of TN_1 with $P_0 = g_P^{-1}(P_2) \subseteq P_1$ and $T_0 = g_T^{-1}(T_2) \subseteq T_1$ with inclusion $f : TN_0 \rightarrow TN_1$.

Remark 4 (Pullback of Timed P/T Nets) In [LGM12] a slightly more general restriction construction is defined, where the inclusion f' is replaced by a time-strict injective morphism, leading also to a time-strict injective morphism f . Moreover, it is shown that the restriction construction is also a pullback construction. This means that given a restriction (1) in Figure 3a of timed P/T nets, then (1) commutes and satisfies the following universal property: For all timed P/T nets TN_4 and timed P/T morphisms $h : TN_4 \rightarrow TN_1$, $k : TN_4 \rightarrow TN_2$ with $g' \circ h = f' \circ k$ there exists a unique timed P/T morphism $m : TN_4 \rightarrow TN_0$ with $f \circ m = h$ and $g \circ m = k$.

Example 5 (Gluing and Restriction of Timed P/T Nets) Figure 4 shows the gluing of timed P/T nets TN_1, TN_2 over the interface TN_0 . The morphism f is an inclusion, whereas g maps the places p_1 and p_2 non-injectively to place $p_{1,2}$, and f_2 is not time-strict because arc-inscription 10 between t_1 and p_1 is mapped to value 5. The gluing object TN_3 contains all elements of TN_1 and TN_2 which are glued together at transition t_1 and places p_1 and p_2 respectively $p_{1,2}$. Vice versa, net TN_0 and morphism g can be considered as restriction of TN_1 and g' along inclusion f' . Net TN_0 contains only those parts of TN_1 that have a corresponding preimage in TN_2 .

Based on these definitions of structuring techniques for timed P/T nets, we can show that our category **TPTNets** fulfills the properties of \mathcal{M} -adhesive systems [EGH10], a general categorical framework for structuring and transforming high-level algebraic structures that generalises the concept of adhesive categories [LS04]. A category \mathbf{C} together with a class \mathcal{M} of monomorphisms closed under composition and isomorphisms is called an \mathcal{M} -adhesive category if \mathbf{C} has pushouts and pullbacks along \mathcal{M} -morphisms, \mathcal{M} -morphisms are closed under pushouts and pullbacks, and pushouts along \mathcal{M} -morphisms satisfy the so-called (vertical) van Kampen property (for more details see [EGH10]).

Theorem 2 (Timed P/T Nets Are \mathcal{M} -Adhesive) *The category $(\mathbf{TPTNets}, \mathcal{M}_{strict})$ is an \mathcal{M} -adhesive category, where $\mathcal{M}_{strict} = \{f = (f_P, f_T) \mid f \text{ is time-strict and } f_P, f_T \text{ are injective}\}$.*

Proof Idea. According to Remarks 3 and 4, the category **TPTNets** has pushouts and pullbacks along time-strict injective morphisms which can be constructed as gluings and restrictions, respectively. The satisfaction of the vertical van Kampen property is shown explicitly in [LGM12]. \square

The fact that our approach of timed P/T nets fits into the abstract categorical framework of \mathcal{M} -adhesive categories means that it is a suitable category for rule-based transformation in the sense of algebraic graph transformation in the double pushout (DPO) approach [EEPT06]. From the results in [BEGG10] we know that there exist so-called initial pushouts for all finite timed P/T nets, enabling us to define a sufficient and necessary condition for the existence of DPO transformations. Moreover, since all results for weak adhesive HLR categories in [EEPT06] are also valid for the slightly more general framework of \mathcal{M} -adhesive categories, we have unique results of direct transformations of timed P/T nets, as well as various analysis techniques for transformations of timed P/T nets, like the concurrency, local Church-Rosser, local confluence, embedding and extension theorems. In the following, we present an example of a timed P/T net transformation rule which we will apply to the timed P/T net in Figure 1.

Example 6 (Timed P/T Net Transformation Rule) The transformation rule $changeRouter = (L \leftarrow K \rightarrow R)$ in Figure 5 moves a client place to another router. The time values are given as parameters in order to allow arbitrary time values to be used. Note that the rule morphisms l and r are \mathcal{M}_{strict} -morphisms.

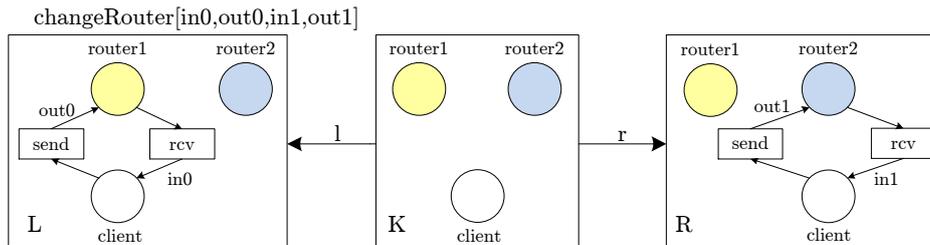


Figure 5: Transformation Rule $changeRouter$

In the DPO diagram in Figure 6, rule $changeRouter$ is applied with $router1$ mapped onto $router1$, $router2$ mapped onto $router4$ and the place $client$ mapped onto $fastclient$ by the injective rule match. This results in a new timed P/T net $Network'$ where the client has been moved from $router1$ to $router4$. In this case, we decided to keep the time values (i. e. $in0 = in1 = 20$, $out0 = out1 = 20$).

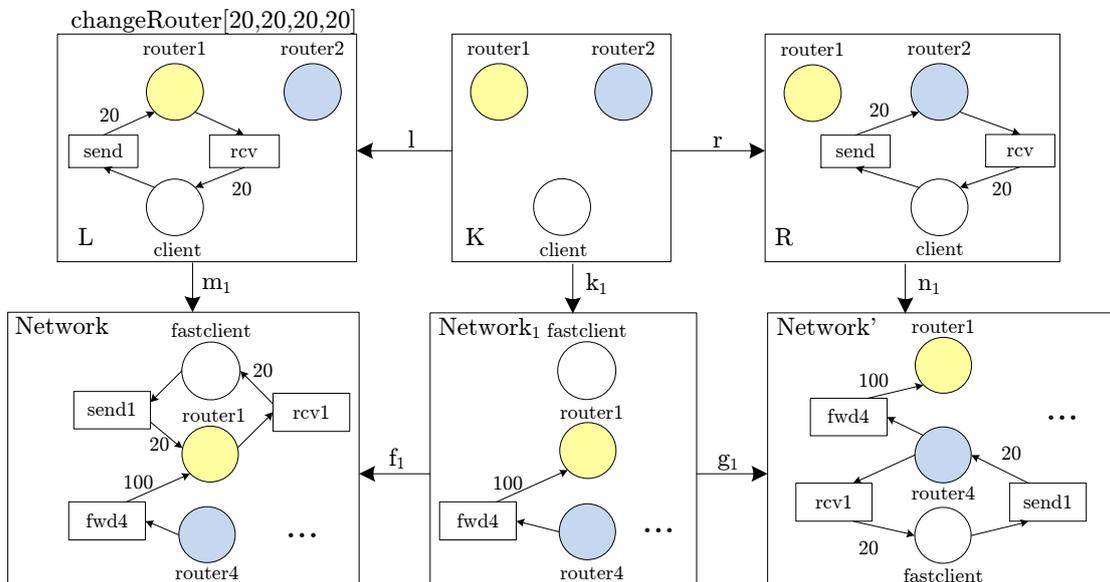


Figure 6: Application of rule $changeRouter$ (injective matching)

Figure 7 shows the rule to be applied with both $router1$ and $router2$ mapped onto $router1$ (non-injectively) and $client$ mapped onto $fastclient$. Hence, in the timed P/T net $Network''$ the

client is not moved, however we chose $out1 = in1 = 5$, resulting in the client being “upgraded” in speed, reducing the time delays of the transitions even further.

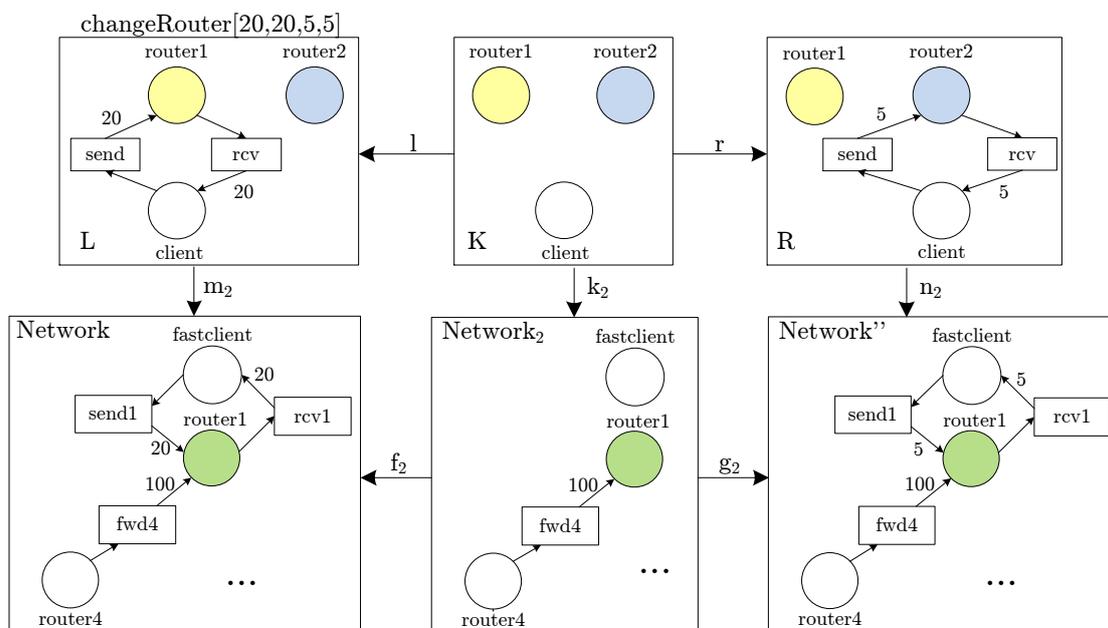


Figure 7: Application of rule *changeRouter* (non-injective matching)

7 Conclusion and Outlook

In this paper we presented a new algebraic approach of timed P/T nets, where transitions have time durations and tokens consist of timestamps. This allows for the modelling of systems with concurrent and time-dependent behaviour as we have shown for our running example presented in Section 3. Due to the algebraic formalisation we are able to define morphisms between timed P/T nets which preserve the firing behaviour as shown in our first main result in Section 5. This enables us to define structuring techniques like the gluing and restriction of timed P/T nets presented in Section 6, leading to an \mathcal{M} -adhesive category, which means that our approach is also suitable for rule-based transformation (also called reconfiguration) of timed P/T nets in the sense of graph transformation [EEPT06].

In future work we will study how to combine our approach of timed nets with the approach of Algebraic High-Level (AHL) nets to obtain a Petri net class of timed AHL nets in order to enable the modelling of time-dependencies and data in an integrated model. Tool support for timed P/T nets and AHL nets is planned as extension of the environment for reconfigurable AHL nets that is currently being implemented in our group [EFS11]. Another topic for future work is the definition of processes of timed P/T nets similar to processes of untimed P/T nets and AHL nets [Ehr04], in order to allow for analysis and optimization of processes regarding not only the concurrent firing semantics but also the time.

Bibliography

- [BD91] B. Berthomieu, M. Diaz. Modeling and Verification of Time Dependent Systems Using Time Petri Nets. *IEEE Transactions on Software Engineering* 17, 1991.
- [BEGG10] B. Braatz, H. Ehrig, K. Gabriel, U. Golas. Finitary M-Adhesive Categories. In Ehrig et al. (eds.), *Proceedings of Intern. Conf. on Graph Transformation (ICGT' 10)*. Volume 6372, pp. 234–249. 2010.
- [BH07] M. Z. Branislav Hruz. *Modeling and Control of Discrete-event Dynamic Systems: with Petri Nets and Other Tools*. Springer, 2007.
- [EEPT06] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theor. Comp. Science. Springer, 2006.
- [EFS11] C. Ermel, W. Fischer, S. Shareef. RONs Revisited: General Approach and Implementation of Reconfigurable Object Nets based on Algebraic High-Level Nets. *ECEASST* 40, 2011.
- [EG11] H. Ehrig, K. Gabriel. Transformation of Algebraic High-Level Nets and Amalgamation of Processes with Applications to Communication Platforms. *International Journal of Software and Informatics* 5(1):207 – 229, 2011.
- [EGH10] H. Ehrig, U. Golas, F. Hermann. Categorical Frameworks for Graph Transformations and HLR Systems based on the DPO Approach. *Bulletin of the EATCS* 102:111–121, 2010.
- [EHP⁺02] H. Ehrig, K. Hoffmann, J. Padberg, P. Baldan, R. Heckel. High-level net processes. In Brauer et al. (eds.), *Formal and natural computing*. Pp. 191–219. Springer, 2002.
- [Ehr04] H. Ehrig. Behaviour and Instantiation of High-Level Petri Net Processes. *Fundamenta Informaticae* 64:1–37, 2004.
- [EM96] C. Ermel, A. Martini. A Taste of Categorical Petri Nets. Technical report 96-9, TU Berlin, 1996.
- [EPR94] H. Ehrig, J. Padberg, L. Ribeiro. Algebraic High-Level Nets: Petri Nets Revisited. In *Recent Trends in Data Type Specification*. Volume 785, pp. 188–206. 1994.
- [ER97] H. Ehrig, W. Reisig. An Algebraic View on Petri Nets. *Bulletin of the EATCS* 61:52–58, February 1997.
- [GLMR05] G. Gardey, D. Lime, M. Magnin, O. H. Roux. Romo: A Tool for Analyzing time Petri nets. In *Proc. CAV'05*. LNCS 3576, pp. 418–423. Springer, 2005.
- [GMMP91] C. Ghezzi, D. Mandrioli, S. Morasca, M. Pezzè. A Unified High-Level Petri Net Formalism for Time-Critical Systems. *IEEE Trans. Softw. Eng.* 17:160–172, 1991.

- [GVH04] S. Gyapay, D. Varró, R. Heckel. Graph Transformation with Time. *Fundamenta Informaticae* 58:1–22, 2004.
- [Jen92] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Volume 1: Basic Concepts. Springer, 1992.
- [JK09] K. Jensen, L. M. Kristensen. *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*. Springer, 2009.
- [JKW07] K. Jensen, L. M. Kristensen, L. Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *Int. Journal on Software Tools for Technology Transfer*, 2007.
- [LGM12] P. Lingnau, K. Gabriel, T. Modica. An Algebraic Approach to Timed Petri Nets with Applications to Communication Networks – Extended Version. Technical report 2012/02, Technische Universität Berlin, 2012.
- [LS04] S. Lack, P. Sobociński. Adhesive Categories. In *Proc. FOSSACS 2004*. LNCS 2987, pp. 273–288. Springer, 2004.
- [MGH11] T. Modica, K. Gabriel, K. Hoffmann. Formalization of Petri Nets with Individual Tokens as Basis for DPO Net Transformations. In Ehrig et al. (eds.), *Proc. Workshop on Petri Nets and Graph Transformation (PNGT'10)*. Volume 40. EASST, 2011.
- [MM90] J. Meseguer, U. Montanari. Petri Nets Are Monoids. *Information and Computation* 88(2):105–155s, 1990.
- [PEHP08] U. Prange, H. Ehrig, K. Hoffman, J. Padberg. Transformations in Reconfigurable Place/Transition Systems. In Degano et al. (eds.), *Concurrency, Graphs and Models: Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*. Volume 5065, pp. 96–113. Springer, 2008.
- [Pet62] C. A. Petri. *Kommunikation mit Automaten*. Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.
- [Rei85] W. Reisig. *Petri Nets: An Introduction*. EATCS Monographs on Theoretical Computer Science 4. Springer, 1985.
- [Rei10] W. Reisig. *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. Vieweg+Teubner, 2010.