Hierarchical Timed High Level Nets and their Branching Processes

Fleischhack, H. and Pelz, E.

Technical Report TR-2002-19
Hierarchical Timed High Level Nets and their Branching Processes

Theoretical paper

Hans Fleischhack
Fachbereich Informatik
Carl-von-Ossietzky – Universität Oldenburg
D-26111 Oldenburg
fleischhack@informatik.uni-oldenburg.de

Elisabeth Pelz
LACL, Faculté de Sciences
Université Paris XII–Val de Marne
61 avenue du Général de Gaulle
F-94010 Créteil Cedex
pelz@univ-paris12.fr

Abstract
The paper aims at defining hierarchical time extensions of M-nets, a fully compositional class of high-level Petri nets. As a starting point, the class of classical timed M-nets are considered, where time intervals of duration are attached to each transition. This class is enriched by two new operations: timed refinement (which extends the class) and hierarchical scoping (which is shown to be a powerful feature for abstraction). It is argued that hierarchical timed M-nets allow for a nice way of designing real-time systems in a top down manner. Moreover, a partial order semantics of hierarchical timed M-nets is defined based on branching processes. The definition is given directly for high level nets, without preliminary unfolding to low level nets. This semantics enables partial order model checking of hierarchical timed M-nets e.g. within the PEP-system.

Keywords: Timed and stochastic nets, partial order semantics

1 Introduction

In the last years time extensions of Petri nets have become more and more studied, see for instance the former volumes of this Petri Nets Conference. We can distinguish more classical approaches as time nets (cf. e.g. [27]) and timed nets (cf. e.g. [29,31]), where different kinds of timing constraints are explicitly added to certain net elements [1–3, 10, 24, 25, 28] from a quite different approach, considering causal time. This approach was first mentioned by [14,30], and has

1 This work has been partially supported by the Procope project PORTA (Partial Order Real Time Semantics).
recently been applied to the M-net Calculus [21–23], a fully compositional class of high-level Petri nets [9, 8]. In this paper, we will contribute to the classical approach. More specifically, we will consider timed nets, where the duration of each action is given by an interval of [earliest time, latest time].

Some authors have already considered time(d) Petri nets in the context of PBC, the low level version of the M-net algebra [1, 10, 18, 24]. We present them briefly in section 5.

What we would like to embody in this paper is a kind of global time constraint. Consider e.g. the requirement preparing a meal should take no more time than one hour. This could be implemented by a group of cooks, whose different manipulations of preparing dishes all have some particular duration. Only those implementations are acceptable, which respect the total limit of one hour. Or, as still a simpler problem consider the following sequence of two timed transitions in figure 1:

![Fig. 1. Sequence of timed transitions](image)

How would you model a constraint saying the consecutive firing of $t_1$ and $t_2$ consumes exactly 100 seconds in total? How would you express this constraint within a timed Petri net? To which parts of the net constraints should be imposed?

In PBC and M-net Calculus, one powerful operation is that of general transition refinement [5, 12, 13]. As a first solution to the above problem one could associate the time interval $[100, 100]$ to a transition $t_1$ which is then refined by the net in figure 1, i.e. by the sequence of $t_1$ and $t_2$. Unfortunately, the semantics of an M-net containing refinements yields a flat net, where all refined transitions have been replaced by appropriate sub-nets. So, for the above problem, a timed extension of general refinement should result in a flat net containing at least 101 alternative sequences of incarnations of $t_1$ and $t_2$, one for every allowed combination of firing times. Obviously, such an unfolding of time is no desirable solution.

Let us illustrate this by a second, a bit more complicated, example:

![Fig. 2. Second sequence of timed transitions](image)
Once more, the execution of the net of figure 2 should not consume more than 100 seconds in total. Now, the flat net with the same timed behavior, would have a branch for each combination of values $c_1$, $c_2$, $c_3$, $c_4$ such that $c_1 + c_2 + c_3 + c_4 \leq 100$ and $c_1$, $c_3$ stand for the time consumed by transitions $t_1$ and $t_3$, $c_2$ for the number of times transition $t_2$ fires (consuming each time exactly one second) and $c_4$ indicating the number of 'tics' (or seconds) after the end of the firing of $t_3$. And, there are also some branches for executions with time-outs. Thus the equivalent flat net may look like the following net, where we only show some possible branches.

![Diagram](image)

**Fig. 3. An equivalent flat net**

The time-constants in figure 3 are as follows: $c_1 + c_2 + c_3 \leq 100$, $c'_1 + c'_2 = 100$ and so the second branch ends by timeout, $c_1 + c_2 \leq 100$, and the right most branch ends by timeout too, after $t_3^y$ having consumed exactly 100 seconds.

Would it be possible to obtain such a flat net by using existing M-net operations, for instance by general refinement where figure 2 is the refining net? Surely not! The reason is that all existing operations only concern the interfaces of nets. Here, as shown in the last figure, the flat net depends on the possible executions: in fact each execution of the net in figure 2 respecting the global
time constraint defines a branch in the flat net of figure 3. This cannot be the result of a composition in the existing M-net algebra.

Hence, in section 2 within the class of timed M-nets (also called duration M-nets) a totally new operation will be introduced, called timed refinement, \( N[\tau_{e,d}] \leftarrow N_1 \). Differing from the composition operators of the M-net calculus, application of timed refinement does not result in a (flat) timed M-net, but will yield a hierarchical net. The meaning of the timed refined transition is that of an additional clock which is visible within the refining net. In general, this operation leads to a tree-like nested structure representing the timely dependencies between the nets involved. The resulting class of nets will be called hierarchical duration M-nets, or hd-M-nets.

In section 3, the new concept of a timed maximal branching process is introduced and – based on this – a partial order semantics for hd-M-nets is defined. At our knowledge, it is the first approach to define directly a branching process semantics for high level nets with time. This concept will allow us to prove properties of hd-nets expressed in a suitable temporal logic with partial order model checking, e.g. within the PEP-system [10,16,17,26].

To facilitate even more a top down development, an other original operation, called hierarchical scoping, will be added in section 4. It will allow to synchronize two transitions (for instance, a sender with a receiver) which will both be timed refined later on (here, by specific fax-machines) over abstract action symbols. Hierarchically scoped hd-nets will become usual (also called concrete) hd-nets after executing the scoping. So hierarchical scoping is a powerful device for abstraction, but no extension of the net class. The fax example will illustrate this later.

Some related work will be discussed in section 5, and in section 6, some conclusions are drawn.

2 Hierarchical duration M-nets

In this section, the notion of a hierarchical duration M-net is introduced and their transition rule is defined. Finally, the definitions are illustrated by means of an example. In the following, we consider only safe M-nets, in the sense that, for every M-net \( N \) and for every reachable marking \( M \), \( M(s) \) is a set for each \( s \in S_N \).

2.1 Basic definitions

A duration M-net is an M-net (cf. appendix ??, where all standard definitions on M-nets can be found), in which the firing of events may consume time. The set of allowed firing times is given by an interval of nonnegative real numbers, the interval having rational bounds, but, following the discussion in [32], without loss of generality only integer firing times will be considered here.
Definition 1 (Duration M-net). A duration M-net (d-Mnet for short) consists of an M-Net $\mathcal{M}$ together with an additional inscription

$$\chi : T_e \rightarrow \mathbb{N} \times (\mathbb{N} \cup \{\omega\})$$

of transitions. For a transition $t \in T_e$, $\chi(t) = (\text{eft}(t), \text{ltf}(t))$ denotes the possible range for the duration of an occurrence of $t$. Altogether, for each transition, the inscription of a transition $t$ is of the form $\iota(t) = \alpha(t).\gamma(t).\chi(t)$ (cf. Appendix ??).

For a d-Mnet $\mathcal{M}$,

$$\text{Events}(\mathcal{M}) = \{ t : \sigma | t \in T \text{ and } \sigma \text{ is an enabling binding for } t \}$$

denotes the set of possible firing events.

A firing event $t : \sigma$ of a d-Mnet $\mathcal{M}$ is enabled if it is so in the underlying M-net, i.e. iff $\sigma$ is an enabling binding of the variables occurring in the scope of $t$. After becoming enabled, $t : \sigma$ starts firing immediately, by removing the input tokens from its preplaces (unless it is prevented from doing so by a conflicting firing event). After that, $t : \sigma$ stays firing for some time delay $\delta$ s.t. $\text{eft}(t) \leq \delta \leq \text{ltf}(t)$. After that, $t : \sigma$ ceases to fire by delivering the output tokens to its postplaces.

Hence, for the definition of the transition rule for a d-Mnet, we have to consider three types of events, namely

1. startfire events, which remove tokens from the preplaces of a transition,
2. endfire events, which put tokens on the postplaces, and
3. tic events, which model passing of time.

So, in a d-Mnet, firing of an event does not only depend on markings but also has to take into account the timing constraints. Therefore a notion of state is introduced which consists of a marking part and a part keeping track of the relevant timing information: A state of a d-Mnet $\mathcal{M}$ is a pair $(M, l)$, consisting of a marking $M$ of $\mathcal{M}$ and a mapping $l$. For each firing event $t : \sigma \in \text{Events}(\mathcal{M})$, the clock vector $l$ keeps track of the amount of time elapsed since $t : \sigma$ started to occur. Hence, $l$ is a partial function, mapping each currently occurring firing event to a natural number. The initial state $S_0 = (M_0, l_0)$ of $\mathcal{M}$ is given by the standard initial marking $M_0$ of $\mathcal{M}$ and the totally undefined clock vector $l_0$.

The formal definition of the transition rule for d-Mnets is given by the case of not timed refined transitions in definition 3.

In a hierarchical d-Mnet $\mathcal{N}_1$, a transition $t$ may be equipped with another (hierarchical) d-Mnet $\mathcal{N}_2$, in which case $t$ is called timed refined by $\mathcal{N}_2$. The intended meaning is that if an event $t : \sigma$ starts firing, a fresh copy $\mathcal{N}_{t, \sigma}$ of $\mathcal{N}_2$ is created and put in parallel with $\mathcal{N}_1$. Concurrently with $\mathcal{N}_1$, $\mathcal{N}_{t, \sigma}$ starts firing in its initial state $S_0$. If an endfire-event of $t : \sigma$ occurs, $\mathcal{N}_{t, \sigma}$ is removed immediately. So, in some sense $t : \sigma$ acts as an additional global clock for $\mathcal{N}_2$. This kind of semantics is called timeout semantics. Alternatively, $t : \sigma$ can be seen as specification of a time restriction which $\mathcal{N}_{t, \sigma}$ has to meet, leading to a
deadlock if it fails to do so. This type of semantics is called *deadlock semantics*. Of course, time refinements can be nested. These considerations lead to the following definition:

**Definition 2 (Hierarchical duration M-net).**

The set $DN$ of hierarchical duration $M$-nets (hd-$M$nets for short) is defined inductively as follows:

- Every $d$-$M$net $N$ is an hd-$M$net. The transitions of $N$ are called not timed refined.
- If $N_1, N_2$ are hd-nets and $t \in T_1$ such that $t$ is not timed refined, then $N = N_1[t \leftarrow N_2]$ is an hd-$M$net. ($N$ is called timed refinement of $t$ in $N_1$ by $N_2$ and $t$ is called timed refined (by $N_1$)).

An inscription of a clock in the graphical representation of a transition $t$ indicates that $t$ is timed refined.

For an hd-$M$net $N$, $CON(N)$ denotes the set of constituent nets, defined by

$$CON(N) = \begin{cases} \{N\} & \text{if } N \text{ is a d-Mnet} \\ CON(N_1) \cup CON(N_2) & \text{if } N = N_1[t \leftarrow N_2] \end{cases}$$

Slightly abusing notation, for $N$,

$$S_N = \bigcup_{N' \in CON(N)} S_{N'}$$

will denote the set of all places of all constituent nets of $N$. Similar notation is used for the set of transitions of $N$ etc.

Moreover,

$$Events(N) = \bigcup_{N' \in CON(N)} Events(N')$$

denotes the set of possible firing events of $N$ and

$$Events_{ref}(N) = \{t : \sigma | t : \sigma \in Events(N) \text{ and } t \text{ is timed refined in } N\}$$

denotes the set of possible timed refined firing events.

For $t : \sigma \in Events_{ref}(N)$ such that $t$ is timed refined by $N'$, let $N_{t, \sigma}$ denote a fresh copy\(^3\) of $N'$, $S_{N_{t, \sigma}}$ its places, etc. Then all places in the refining nets are defined as

$$S_{ref}(N) = \bigcup_{t : \sigma \in Events_{ref}(N)} S_{N_{t, \sigma}}.$$

---

\(^3\) We assume that different occurrences of a firing event are distinguished by different names of the bindings. Hence, a single copy of the timed refining net for each timed refined firing event is sufficient for the following definitions.
A marking $M$ of $N$ is a set over $S_N$ such that, for all $s$, $M_s \in \mathcal{P}_f(\tau(s))$. And

$$\text{Enabled}(M) = \{ t : \sigma \in \text{Events}(N) \text{ and } M \text{ activates } t \text{ in mode } \sigma \}$$

denotes the set of firing events of $N$ which can occur at a marking $M$.

$St = (M, I)$ is a state of $N$, iff

- $M$ is a marking of $N$ and
- $I : \text{Events}_{ref}(N) \rightarrow \mathbb{N} \cup \{\bot\}$, such that $I(t : \sigma) \neq \bot$ implies $I(t : \sigma) \leq \mathit{lft}(t)$.

Assume that $N$ is of the form

$$N = (\ldots (N_0[t_1 \leftarrow N_1])[t_2 \leftarrow N_2] \ldots [t_n \leftarrow N_n]),$$

where possibly $n = 0$. Then $St_0(N) = (M_0, I_0)$ such that

- $M_0 = M^*(N_0)$ is the standard initial marking\(^3\) and
- $I(t : \sigma) = \bot$ for all $t : \sigma \in \text{Events}(N)$

is called the initial state of $N$.

\section{The transition rule}

As for d-Mnets, for the transition rule of an hd-Mnet we distinguish three types of events, namely

1. \textit{Startfire events}: A startfire event (denoted as $[t : \sigma]$) has to occur immediately if $t : \sigma$ becomes enabled in the underlying M-net. Upon occurrence of $[t : \sigma]$, the input tokens are removed from its preplaces, the clock associated to $t : \sigma$ is started, and $t : \sigma$ is called active. Moreover, if $t$ is timed refined, the net associated to $t : \sigma$ is set to its initial marking.

2. \textit{Endfire events}: For the occurrence of an endfire event (denoted as $t : \sigma$) we have to distinguish between transitions which are timed refined and those which are not.

   - If $t$ is not timed refined, then $t : \sigma$ may occur if the clock associated to $t : \sigma$ shows a time $\delta$ such that $\mathit{eft}(t) \leq \delta \leq \mathit{lft}(t)$. In this case, upon occurrence of $t : \sigma$, the output tokens are delivered at the postplaces of $t$ and the clock associated to $t : \sigma$ is reset.
   - Otherwise, i.e. if a net $N_1$ is associated to $t$, $t : \sigma$ may occur if either the copy of $N_1$ associated to $t : \sigma$ has reached its final marking and the clock associated to $t : \sigma$ shows a time $\delta$ such that $\mathit{eft}(t) \leq \delta \leq \mathit{lft}(t)$ or if the latest firing time is reached, i.e. if $\delta = \mathit{lft}(t)$.

For the effect of the occurrence of $t : \sigma$ two different approaches are considered:

\(^3\) $M^*(N)$, the standard initial marking of $N$ is given by $M^*(s) = \tau(s)$ for every entry place $s$ and $M^*(s) = \emptyset$ for all other places.
- deadlock semantics: The net $N$ (and all other nets of $CON(N)$) are stopped by removing all tokens from all places.
- timeout semantics: The net $N_t,\sigma$ is stopped by removing all tokens from places in $S_{N_t,\sigma}$

3. Tic events: A tic event (denoted as $\checkmark$) is enabled iff there is no firing event which must either start firing or stop firing. Upon occurring, a tic event increments the clocks for all active firing events. Hence, tic events are global.

We write $S \xrightarrow{e} S'$ to denote that event $e$ is enabled at state $S$ and $S \xrightarrow{e} S'$ to denote that $S \xrightarrow{e}$ and that the occurrence of $e$ at $S$ leads to state $S'$.

**Definition 3 (Transition rule).**

Let $N$ be an hd-Mnet.

1. **startfire events:**
   - $(M, I) \xrightarrow{t : \sigma}$ iff $t : \sigma \in Events(N)$ and $M$ activates $t$ in mode $\sigma$.
   - $(M, I) \xrightarrow{(M', I')}$ iff $(M, I) \xrightarrow{t : \sigma}$ and
     - **Case 1:** $t$ is not timed refined. Then
       - $M' = M - \bigcup_{t \in T} \{ t(s, t) | [\sigma] \}$
       - $I'(t' : \sigma') = \begin{cases} 0 & \text{if } t = t' \text{ and } \sigma = \sigma' \\ I'(t' : \sigma') & \text{otherwise} \end{cases}$
     - **Case 2:** $N = N_1 [t \leftarrow N_2]$. Then
       - $M' = M - \bigcup_{t \in T} \{ t(s, t) | [\sigma] \} \cup \bigcup_{t \in T} \{ t(s, t) | [\sigma] \}$
       - $I'(t' : \sigma') = \begin{cases} 0 & \text{if } t = t' \text{ and } \sigma = \sigma' \\ \bot & \text{if } t' : \sigma' \in Events(N_t, \sigma) \\ I'(t' : \sigma') & \text{otherwise} \end{cases}$

2. **endfire events:**
   - **Case 1:** $t$ is not timed refined. Then
     - $(M, I) \xrightarrow{t : \sigma}$ iff $\text{eft}(t) \leq I(t : \sigma) \leq \text{lf}(t)$.
     - $(M, I) \xrightarrow{(M', I')}$ iff $(M, I) \xrightarrow{t : \sigma}$ and
       - $M' = M + \bigcup_{t \in T} \{ t(s, t) | [\sigma] \}$
       - $I'(t' : \sigma') = \begin{cases} \bot & \text{if } t = t' \text{ and } \sigma = \sigma' \\ I'(t' : \sigma') & \text{otherwise} \end{cases}$
   - **Case 2:** $N = N_1 [t \leftarrow N_2]$. Then
     - $(M, I) \xrightarrow{t : \sigma}$ iff $\text{eft}(t) \leq I(t : \sigma) \leq \text{lf}(t)$
     - $(M, I) \xrightarrow{(M', I')}$ iff $(M, I) \xrightarrow{t : \sigma}$ and
       - $I'(t' : \sigma') = \begin{cases} \bot & \text{if } t = t' \text{ and } \sigma = \sigma' \\ I'(t' : \sigma') & \text{otherwise} \end{cases}$
(a) **deadlock semantics:**
* $M' = \emptyset$
* $I'(t' : \sigma') = \bot$ for all $t' : \sigma' \in \text{Events}(N)$.

(b) **time-out semantics:**
* $M' = M + (\bigcup_{s \in T} t(s)[\sigma] - M(N_{t,\sigma})$
* $I'(t' : \sigma') = \begin{cases} \bot & \text{if } (t = t' \text{ and } \sigma = \sigma') \\ I'(t' : \sigma') & \text{otherwise} \end{cases}$

3. **tie events:**
- $(M, I) \xrightarrow{\bot}$ iff
  $\{ t : \sigma \mid t : \sigma \in \text{Events}(N) \text{ and } M \text{ activates } t \text{ in mode } \sigma \} = \emptyset$ and, for all $t : \sigma \in \text{Events}(N)$, $I(t : \sigma) \neq \bot$ implies $I(t : \sigma) < [t](t)$.
- $(M, I) \xrightarrow{I'} (M', I')$ iff $I' = I \oplus 1$, $(M, I) \xrightarrow{I}$ and $M = M'$.
  where $(I \oplus 1)(t : \sigma) = I(t : \sigma) + 1$ for all $t : \sigma$.

All the usual dynamic concepts (occurrence sequences, step sequences, set of reachable markings etc.) follow from the transition rule in the standard way.

### 2.3 An example

The concepts introduced in this section are illustrated by means of an example. Consider a simple fax machine, which may either send or receive a fax\(^4\). Figure 4 shows an hd-Mnet $N_0$ modelling an abstract view of the fax. We assume that $\text{fax}$ is an abstract action symbol, that $\alpha(t_{\text{send}}) = \{\text{fax}\}$, and that $\alpha(t_{\text{receive}}) = \{\overline{\text{fax}}\}$. Hence, $t_{\text{send}}$ as well as $t_{\text{receive}}$ may be timed refined transitions.

![Fig. 4. A hd-Mnet $N_0$ modelling a simple fax machine](image)

Figure 5 shows a timed refinement $N_1$ of transition $t_{\text{send}}$ of the net $N_0$: After dialing (which consumes one time unit) the fax either gets connected (consuming 1 to 3 time units) and may send its data (consuming 1 to 5 time units), or it is not connected (which consumes one time unit) and has to wait for 20 time units and then has to dial again and so on. Note, that $N_1$ may unsuccessfully try to establish a connection for an unbounded amount of time. The timing restrictions of $t_{\text{send}}$, however, will stop $N_1$ after at most 100 time units.

\(^4\) $M^\tau$ denotes the standard final marking of an M-net $N$ and is given by $M^\tau(s) = \tau(s)$ for every exit place $s$ and $M^\tau(s) = \emptyset$ for any other place.

\(^5\) Note, that by no means we try to model a real fax machine. This is just to illustrate the definitions.
Figure 6 shows an (oversimplified) view of the receive action $t_{\text{receive}}$ by an hd-Mnet $N_2$. Hence altogether the fax is described by an hd-Mnet $N$ such that

$$N = N_0[t_{\text{send}} \leftarrow N_1][t_{\text{receive}} \leftarrow N_2].$$

3 Partial order semantics of hierarchical duration M-nets

In this section, the new notion of branching process of a hierarchical duration net is introduced. As for a P/T-net the maximal branching process $\beta_m$ associates a partial order semantics to each safe hd-Mnet $N$ [16]. Moreover, following the line of [16], a finite representation of $\beta_m$ can be defined, which in turn can be used to analyse qualitative temporal properties of $N$, e.g. using the PEP-system.

A causal process describes a possible run of $N$, displaying the causal dependencies of the events that take place during the run. A branching process can represent several alternative runs of $N$ in one structure and hence may be seen as the union of some causal processes. Therefore, the maximal branching process represents all causal process within a single structure. A branching process consists of an occurrence net and a homomorphism.

An occurrence net ON = $(B, E, G)$ is an acyclic net such that $|\ast b| \leq 1$ for all $b \in B$, no $e \in E$ is in self conflict and, for all $x \in (B \cup E)$, the set of elements $y \in (B \cup E)$ such that $x \leq y$ is finite, where $\leq$ refers to the partial order induced by $G$ on $B \cup E$. Nodes $x_1, x_2$ are in conflict ($x_1 \not\equiv x_2$) if there are distinct events
$e_1, e_2 \in E$ such that $e_1 \cap e_2 \neq \emptyset$ and $(e_1, x_1), (e_2, x_2) \in G^*$. A node $x \in (B \cup E)$ is in self conflict if $x \notin x$. The elements of $B$ and $E$ are called conditions and events, respectively. $\text{MIN}(ON)$ denotes the set of minimal elements of $ON$ w.r.t. $\leq$.

The homomorphism is used to connect conditions and events of an occurrence net to places and transitions of the net whose behaviour is described. Here, in addition, we have to define some labels representing the time part. So let $N$ be an hd-Met. To represent the clock vector part of a state, we introduce new place labels $(t : \sigma, i)$ with $i \in N$ for any firing event $t : \sigma \in \text{Events}(N)$. The intended meaning is that $(t : \sigma, i)$ indicates that event $t : \sigma$ is firing since $i$ time units. The set of these clock labels is called $St(N)$. Thus a clock vector can be represented as part of a marking.

Next, we give a low level representation of a marking: Token $v$ at place $s$ is represented by a label $s_v$, and we let

$$S_D(N) = \{s_v|s \in (S_{ref}(N) \cup S(N)) \text{ and } v \in \alpha(s)\}$$

denote the set of data labels of $N$. The set of all low level labels is then given by

$$S^H(N) = S_D(N) \cup S_I(N).$$

Moreover, define

$$\phi : \text{States}(N) \rightarrow \mathcal{M}_I(S^H(N))$$

by

$$\phi(M, I)(x) = \begin{cases} M(s)(v) & \text{if } x = s_v \in S_D(N) \\ 1 & \text{if } x = (t : \sigma, i) \in S_I(N) \\ 0 & \text{otherwise.} \end{cases}$$

So, for every state $(M, I)$ of $N$, $\phi$ computes a multiset of labels in $S^H(N)$ (which in fact is a set if $N$ is safe). Note that $\phi$ is a partial bijection.

The set $\mathcal{F}(N)$ of firing elements of $N$ is defined by

- $\text{Tic}(N) = \{\checkmark_C|C \subseteq S_I(N) \text{ and } (t : \sigma, i) \in C \text{ implies } i < \text{left}(t)\}$
- $\text{Startfire}(N) = \{(t : \sigma | t : \sigma \in \text{Events}(N)\}$
- $\text{Endfire}(N) = \{t : \sigma|_{(M, I)} | t : \sigma \in \text{Events}(N) \text{ and } (M, I) \in \text{States}(N) \text{ and } \text{left}(t) \leq I(t : \sigma) \leq \text{left}(t)\}$
- $\mathcal{F}(N) = \text{Tic}(N) \cup \text{Startfire}(N) \cup \text{Endfire}(N)$.

Let $fe \in \mathcal{F}(N)$ and define

$$\text{pre}(fe) = \begin{cases} C & \text{if } fe = \checkmark_C \\ \phi(\bigcup_{s \in S} t(s, t)[\sigma]) & \text{if } fe = [t : \sigma] \\ \{(t : \sigma, I(t : \sigma))\} & \text{if } fe = t : \sigma|_{(M, I)} \text{ and } t \text{ not refined} \\ \{(t : \sigma, I(t : \sigma))\} \cup \phi(M, I)[CON(N, t)]^7 & \text{if } fe = t : \sigma|_{(M, I)} \text{ and } t \text{ refined.} \end{cases}$$

Note, that if we are dealing with safe nets, there can at most be one entry $(t : \sigma, i)$ for any firing event $t : \sigma$. 

\[\]
and

\[
\text{post}(fe) = \begin{cases} 
(t : \sigma, i + 1) & \text{if } fe = \sqrt{C} \\
(t : \sigma, 0) & \text{if } fe = [t : \sigma] \text{ and } t \text{ is not refined} \\
(t : \sigma, 0) \cup \phi(S_0(N_{t,\sigma})) & \text{if } fe = [t : \sigma] \text{ and } t \text{ is refined} \\
\phi(\bigcup_{s \in S} \iota(t, s)[\sigma]) & \text{if } fe = [t : \sigma]_{(M, I)}.
\end{cases}
\]

Now a homomorphism from an occurrence net ON to an hd-Mnet N is defined to be a mapping

\[
\pi : B \cup E \to (\mathcal{S}(N) \cup \mathcal{F}(N))
\]
such that

1. \(\pi(B) \subseteq \mathcal{S}(N)\) and \(\pi(E) \subseteq \mathcal{F}(N)\);
2. for all \(e \in E\), the restriction of \(\pi\) to \(e^*\) is a bijection from \(e^*\) to \(\text{pre}(\pi(e))\);
3. for all \(e \in E\), the restriction of \(\pi\) to \(e^*\) is a bijection from \(e^*\) to \(\text{post}(\pi(e))\);
4. the restriction of \(\pi\) to \(\text{MIN}(ON)\) is a bijection from \(\text{MIN}(ON)\) to \(\phi(M^e, I_0)\);
5. for all \(e_1, e_2 \in E\) it holds that if \(e_1^* = e_2^*\) then \(e_1 = e_2\).

The following algorithm nondeterministically constructs the maximal branching process \(\beta_m = (B, E, G, \pi)\) of \(N\) in the following way: If any startfire event may occur, then the algorithm chooses among the enabled startfire and endfire events. Otherwise, if no endfire event is forced to occur, it will choose one of the tic events and the enabled endfire events. Otherwise, it chooses one arbitrary enabled endfire event.

Algorithm Maximal Branching Process:

Let \(E = G = \emptyset\).

Loop

IF Possible(fe) for some fe ∈ Startfire
    THEN
        Choose fe ∈ Startfire(N) ∪ Endfire(N) such that Possible(fe)
    ELSE
        IF (For all fe ∈ Endfire it holds that (not Necessary(fe)))
            THEN
                Choose fe ∈ Tic(N) ∪ Endfire(N) such that Possible(fe)
            ELSE
                Choose fe ∈ Endfire(N) such that Possible(fe)
        FI
    FI

 FOOL

where

\(\text{Denotes the restriction of } \phi(M, I) \text{ to the set of constituent nets of } N_{t,\sigma}.\)
- \( fe \in \mathcal{FE}(N) \) is called Possible iff there exists a conflict free set \( \{b_1, \ldots, b_k\} \subseteq B \) such that \( \pi(\{b_1, \ldots, b_k\}) = \text{pre}(fe) \) and, for all \( e \in E \), \( ((b_i, e) \in G \) for all \( i \leq k \) \( \Rightarrow \pi(e) \neq fe \).
- \( fe \in \text{Endfire}(N) \) is called Necessary iff Possible(\( fe \)) and \( (\pi(b_i) = (t : \sigma, \{f_i(i)\}) \text{ for some } i \in \{1, \ldots, k\} \text{ and some firing event } t : \sigma) \).
- Extend(\( \beta, fe, \beta' \)) iff Possible(\( fe \)) and \( \beta' \) is constructed from \( \beta \) by adding a new event \( e \), new conditions \( b'_1, \ldots, b'_i \), new arcs from \( b_1, \ldots, b_k \) to \( e \), new arcs from \( e \) to \( b'_1, \ldots, b'_i \), and extending \( \pi \) such that \( \pi(e) = fe \) and \( \lambda(\{b'_1, \ldots, b'_i\}) = \text{post}(fe) \).

Figure 7 shows an initial part of the maximal branching process of the hd-Mnet \( N_1 \).

A co-set is a set \( B' \subseteq B \) of conditions of \( \beta_m \) such that, for all \( b \neq b' \in B' \), neither \( b < b' \) nor \( b' < b \) nor \( b \neq b' \). A cut is a maximal co-set \( C \) (w.r.t. set inclusion). The marking \( \text{Mark}(C) \) associated to a cut \( C \) is given by \( \text{Mark}(C) = \{\pi(b) \mid b \in C\} \).

We have the following

**Proposition 1.** Let \( N \) be an hd-Mnet.

A state \( St \) is reachable in \( N \) iff the maximal branching process \( \beta_m \) contains some cut \( C \) such that \( \phi(St) = \text{Mark}(C) \).

4 Operations on hierarchical duration M-nets

In this section, we define composition operations on hd-Mnets which create a new hd-Mnet out of one, two, or three given hd-Mnets. We consider operators of

---

Footnote: If \( fe \in \text{TIC}(N), \{b_1, \ldots, b_k\} \) has to be a maximal conflict free set.
two kinds: those concerning place interfaces and thus the control flow and those concerning transition interfaces and thus capabilities for communication.

The first class of operators consists of

− sequential composition $;$,
− parallel composition $||$,
− choice composition $\sqcup$ and
− iteration composition $[\ast\ast]$.

These are (mutatis mutandis) defined as for M-nets (cf. Appendix ??).

On one hand, the second class of operators contains the classical communication operator $\&$ for scoping based on synchronisation and restriction, which also is essentially the same as for M-nets. Let us just remark that for a new transition $t_{12}$ arising out of $t_1$ and $t_2$ by basic synchronisation, the duration is calculated as follows:

\[
\chi(t_{12}) = (\max(\epsilon ft(t_1), \epsilon ft(t_2)), \min(lft(t_1), lft(t_2))).
\]

On the other hand, we define a new operation called hierarchical timed scoping. This operation can be used to synchronize over timed refined transitions and their respective refining $\text{hd-Mnets}$, thus allowing for a modular hierarchical specification of the timed behaviour of systems. The meaning of hierarchical timed scoping is the following: Given an $\text{hd-Mnet} N$ and transitions $t_1, t_2$ which are timed refined by $\text{hd-Mnets} N_1$ and $N_2$, respectively, such that $t_1$ and $t_2$ have to synchronize with respect to an action symbol $A$. Then, the hierarchical scoping of $t_1$ and $t_2$ (w.r.t. $A$) is given by the synchronized transition $t_{12}$, timed refined by the net which results from the parallel composition of $N_1$ and $N_2$, scoped with respect to the action symbols which $N_1$ and $N_2$ have in common. For the formal definition we have to bear in mind, that $N_1$ and $N_2$ in turn may contain timed refined transitions.

First, we introduce a special type of action symbols: Let $\mathcal{AA} \subseteq A$ such that $\text{arity}(a) = 0$ for all $a \in \mathcal{AA}$. The set $\mathcal{AA}$ is called the set of abstract action symbols. Abstract action symbols are used for the synchronisation of timed refined transitions. The remaining ones, $\mathcal{CA} = A \setminus \mathcal{AA}$ are called concrete. They are used for the (classical) synchronisation of not timed refined transitions. We fix that the action label of a transition $t$ of a $\text{hd-Mnet}$, $\alpha(t)$, is either a (possibly empty) multiset of concrete actions, if $t$ is not timed refined, or a (possibly empty) multiset of abstract actions, if $t$ is timed refined.

Moreover, for an $\text{hd-Mnet} N$ let $\mathcal{CA}(N)$ and $\mathcal{AA}(N)$ denote the set of concrete, respectively abstract action symbols occurring in $N$. A $\text{hd-Mnet}$ is called concrete if all its action labels are in $\mathcal{CA}(N)$ (we refine the usual notion of $\text{hd-Mnets}$, as defined in section 2).

Now, hierarchical scoping $\text{hsc}$ is defined inductively in the following way:

\textbf{Definition 4.} 1. Let $N$ be an $\text{hd-Mnet}$ such that $N$ contains no timed refined transitions. Then the timed hierarchical scoping of $N$ is just the net $N$:

\[
\text{hsc} (N) = N.
\]
2. Let $N$ be an $hd$-Mnet and $t_1, t_2 \in T$ such that $\text{var}(t_1) \cap \text{var}(t_2) = \emptyset$ and $t_1, t_2$ are timed refined, i.e. $N = N_0[t_1 \leftarrow N_1, t_2 \leftarrow N_2]$ for $hd$-Mnets $N_1$ and $N_2$ and, for some $A \in \mathcal{A}^A(N)$, $A \in \alpha(t_1)$ and $\overline{A} \in \alpha(t_2)$. Then the hierarchical basic scoping of $t_1$ and $t_2$ in $N$ w.r.t. $A$ is defined by

$$t_{\text{scbasic}}(N, t_1, t_2, A) =$$

$$N'[t_{12} \leftarrow (N_1 \mid N_2 \text{ s.c. } (\mathcal{C}A(N_1) \cup \mathcal{C}A(N_2))) \textbf{ hsc } (\mathcal{A}^A(N_1) \cup \mathcal{A}^A(N_2)).$$

where $N'$ is given by

- $S_{N'} = S_N$.
- $T_{N'} = T_{N_0} \cup \{t_{12}\} \setminus \{t_1, t_2\}$.
- $\bullet \ t_{12} = t_1 \cup t_2$ and $t_{12}^* = t_1^* \cup t_2^*$.
- $\alpha(t_{12}) = \alpha(t_1) \cup \alpha(t_2) - \{A, \overline{A}\}$.
- $\gamma(t_{12}) = \gamma(t_1) \cup \gamma(t_2)$.
- $\chi(t_{12}) = (\max(\text{eft}(t_1), \text{eft}(t_2)), \min(\text{lift}(t_1), \text{lift}(t_2)))$.

3. Let $N$ be an $hd$-Mnet. Then $\textbf{ hsc } (N) = N \textbf{ hsc } (\mathcal{A}^A(N))$, the timed hierarchical scoping of $N$ w.r.t. all abstract action symbols of $N$, is constructed in two steps:

\textbf{step 1}: Construct $N'$ such that $N'$ is the smallest $hd$-Mnet satisfying

(a) $S = S'$.

(b) Every transition of $N$ (and its surrounding arcs) is also in $N'$, with the same inscriptions as in $N$.

(c) If $t_1$ is a transition of $N$ and $t_2$ is a transition of $N'$ such that one of them contains an abstract action symbol $A \in \mathcal{A}^A(N)$ in its label and the other one the abstract action symbol $\overline{A}$, then any transition $t$ arising through a basic hierarchical scoping over $A$ out of $t_1$ and $t_2$ are also in $N'$, as well as its surrounding arcs.

\textbf{step 2}: Construct $\textbf{ hsc } (N)$ from $N'$ by removing every timed refined transition $t$ with $\alpha(t) \neq \emptyset$ (as well as its surrounding arcs) from $N'$, i.e. $\textbf{ hsc } (N)$ is a concrete $hd$-Mnet.

The following proposition is an immediate consequence of the definition of the hierarchical scoping operator $\textbf{ hsc }$.

**Proposition 2.** Let $N$ be an $hd$-Mnet. Then $\textbf{ hsc } (N)$ is an $hd$-Mnet without any occurrence of abstract actions or hierarchical scoping operators. Thus $\textbf{ hsc } (N)$ is a concrete $hd$-Mnet.

This means that $\textbf{ hsc } (N)$ will just denote an abstract view of a much more detailed net, which is an adequate approach for a top down design of concurrent systems with complicated timing requirements.

Hierarchical scoping may be applied to the $hd$-Mnet $N$ of figure 4, which involves basic hierarchical synchronisation w.r.t. $\text{fax}$ applied to $t_{send}$ and to $t_{receive}$ (which of course is more sensible if $t_{send}$ and $t_{receive}$ belong to different $fax$
machines), which in turn leads to synchronisation of $N_1$ of FaxA (cf. figure 5) and $N_2$ of FaxB (cf. figure 6). The net of figure 8 shows

$$N_1 || N_2 \text{ se } (CA(N_1) \cup CA(N_2)) \text{ hasc } AA(N_1) \cup AA(N_2)$$

$$= N_1 || N_2 \text{ sc } \{\text{connect, send}\} .$$

5 Related Work

Three time extensions of the low level Petri box calculus, but not on the M-net level have been presented at the last Petri Nets Conferences:

Koutny [24] defines a timed Petri box calculus by extending the PBC with waiting times for transitions and studies then its structural operational semantics. This approach corresponds to the time Petri net model and is fairly different from our one chosen in this paper.

In [1] an extension of the PBC with time (as duration of actions) is given and operational as well as denotational semantics are defined, thereby considering all operations of the PBC, including general refinement. The main difference, however, lies in the fact, that they do not consider inherently hierarchical nets and, hence, they also do not consider transitions whose timely behaviour may be constrained by the duration times of several transitions. So, in their approach only local time constraints can be modelled by finite timed Petri nets.

In [18], a method for proving qualitative temporal properties of time Petri nets was presented. As pointed out in section 3, this method can be lifted to hd-Mnets.

6 Conclusion

Starting from the notion of timed Petri net, we have introduced as extension the model of hierarchical duration M-nets (hd-Mnets), which is in fact a double extension: on the one side a passage to compositional high level nets and on the
other side the new possibility to treat global timing constraints for subnets. Resulting from the new operation of timed refinement, the duration of a transition occurrence in an hd-Mnet may be influenced by several hierarchically nested clocks (of the hierarchically superordinate transitions), where the hierarchical nesting is such that the time slots of two different clocks are either contained in one another or their intersection is empty. We have also added an abstract operation called hierarchical scoping which is a powerful device for modular top down development of distributed timed systems. This feature does not extend the expressive power of the class of hd-Mnets.

As an original contribution, we have addressed here for the first time a branching process semantics directly for high level nets with time: Partial order semantics of hd-Mnets have been introduced in terms of their timed maximal branching process to define their concurrent timed behavior. We propose this definition without preliminary unfolding to a low level hd-net: This involves a simultaneous treatment of bindings, choice, clock-counting, time respect (or time-outs).

The semantics so defined allows for proving qualitative temporal properties of systems described by hd-Mnets using already existing tools like the model checking component of the PEP system. Future work will be dedicated to extend the underlying temporal logics to cope also with quantitative temporal properties.

Ongoing work includes the comparison of hd-Mnets, presented here, with causal time M-nets, based on implicit treatment of time constraints (cf. [21–23]). It can be proved that the former can be simulated by the latter w.r.t. concurrent timed behaviour. First results show important differences: for instance, overlapping time requirements can not be modelled by an hd-Mnet, but it can with a causal time M-net [19]. This comparison of the explicit and implicit approach in the treatment of time can be found in [20].

As it is well known, timed Petri nets can be simulated by means of time Petri nets, but not vice versa. It is therefore interesting to define and study the notion of hierarchical time M-nets. The results will be presented within a forthcoming paper where the techniques needed for the translation will be totally different from those used here for timed nets.

References


