Time Petri nets for modelling Civil Litigation

Robert Valette, Brigitte Pradin-Chézalviel

LAAS-CNRS, 7, Av. du Colonel Roche
F-31077 Toulouse Cedex, France
Tel : (+33) 5 61 33 64 09
e-mail : {robert,chezalviel}@laas.fr

1 Introduction

Since the seminal paper of [MH71], other authors [FN97] have suggested to use Petri nets for a formal representation of the temporal structures of legal systems. The objective of this paper is twofold:

- since the paper of J.A. Meldman and A.W. Holt is more than 25 years old, it is necessary to point out what is new in Petri net theory with respect to time constraints expression,
- as Petri net application for legal systems is still marginal, it is also necessary to underline the benefits of a formal specification of such constraints in legal systems.

Indeed, the temporal dimension is important in the legal domain. A lot of rules and regulations include temporal aspects: for example the maximum delay for a property transaction or a trial setting down. The major part of legal rules comprise precedence relations (an action $a_2$ has to be preceded by an action $a_1$). Some actions, as sending a registered mail, have a duration that can be delimited by a time interval (at least one day — three days at most).

The forward chaining of legal operations in procedures and their temporal structures have to be represented by models capable of precisely capturing the concurrency between two parties or the choices between two alternative actions. The fact that Petri nets can be very useful for modelling such systems has already been pointed out in [MH71] in 1971 and more recently in [FN97].

What could be the benefits of a formal model of the temporal structures of juridical rules and procedures? The first one is the possibility of a formal forward-looking consistency analysis. Indeed, when maximal delay constraints are introduced, it is necessary to check if they can be met when the progress of a procedure is normal. It is also useful to point out under which circumstances the maximal delay constraints are violated. The second one, which could be called “monitoring”, is the possibility to analyse, in real-time, if some procedure progress is likely to violate the maximal delay constraints, or not, in the next future.
Before presenting some temporal extensions of Petri nets and exemplifying them by means of the formal temporal specification of a civil litigation problem, let us recall the main characteristics of Petri nets. They consist of a definition of a structure (places, transitions, arcs) and that of a dynamic behaviour (current marking and token player algorithm). Places (denoted as circles) represent partial states while transitions (denoted by bars) stand for asynchronous events and actions. Oriented arcs express relations between actions and states: input places for a transition establish the preconditions of the action associated with it and output places exhibit which partial states are reached after this action. The token flow denotes the dynamic aspect and the current marking represents the current state of the net. The reachability between two markings (is the state denoted by marking $M_2$ reachable from the state denoted by $M_1$ by means of the occurrence of some scenario i.e. by some sequence of actions and events?) is completely defined by the Petri net structure.

To model legal systems some advantages are very useful. Choices can be easily expressed [MH71]; the simplest case is described by two transitions $t_1$ and $t_2$ sharing the same only input place $p$. When the token load of place $p$ is equal to one, only one of these two transitions will be fired. The net structure expresses the two possible paths. It is known that only one of them will occur, but the decision making process (and its possible subjectivity) is let out of the formal model.

Petri nets also clearly express a possible concurrency between two processes. It is possible to independently describe processes and then analyse their combined behaviour which can be concurrent (independent) during some phases of their evolution, in competition for some resources during others (two trials and one audience room) or in co-operation (a plaintiff and his advocate).

With respect to this aspect, Petri nets lead to more powerful models than finite state automata. Automata are very often large ones and concurrency is not explicit because all event occurrences are ordered by a total order relation. On the opposite, a Petri net can represent a partial order relation between event occurrences and therefore explicitly capture concurrency.

This paper is organised as follows. Temporal reasoning with Petri nets is presented at section 2: the different logical approaches are briefly reviewed. Section 3 looks at the different ways to attach explicit time information to Petri nets. They are discussed essentially considering their expressiveness. The $t$-time Petri net model is then used for modelling the civil litigation problem presented at section 4.

## 2 Petri net and temporal reasoning

Temporal reasoning involves considering logical propositions whose truth values vary along time. For example, the proposition “the trial is on” is true between the two events “the trial begins” and “the trial terminates”. These kinds of propositions are not of the same nature as the classical logical propositions such as “humans are mortal” whose truth values are eternal. The monotonicity in classical logic is precisely the property which entails the fact that once a proposition has been proved true, it is true for ever.

In order to cope with this difficulty, temporal logics have been developed. In such
logics the propositions are assumed to be true within some “worlds” and an implicit automaton describes how the event occurrences entail world changes. It has to be pointed out that the logic implication cannot describe the state changes. Consequently, these approaches allow to reason on propositions which are true for some state (world) or for a subset of states, or for all states, but not to reason on sequences of events or on explicit durations of scenarios made up of events and states.

Similarly, the theory of action and time developed by J.F. Allen [Al83, Al84] does not offer a solution. Actually, a large number of relations which can be verified by time intervals during which propositions are true have been defined. But implicitly, these relations are based on a unique common clock and this entails the fact that any two events $e_1$ and $e_2$ can be ordered ($e_1$ is before $e_2$ or after $e_2$) which is contradictory with the explicitation of concurrency. Indeed if two events $e_1$ and $e_2$ are concurrent, this means that no precedence relation can be associated with them.

Ten years ago, a new logic has been defined: linear logic [Gi87]. In this logical system, propositions are consumed and produced by the deductions; indeed, the linear logic implication denotes a state change which entails that propositions (which were available) are no longer available, and that new ones become available. So the changes between two worlds are explicitly expressed. The relationships between this logic and Petri net theory have very quickly been pointed out [Br89, GG89]. This means that it is possible to reason over a Petri net by means of this logic [GPV97] and therefore to cope with concurrency and propositions whose truth value changes in a sound formal framework.

3 Explicit time constraints in Petri nets

Petri nets intrinsically deal with implicit time constraints (qualitative precedence relations) but explicit ones can be expressed by adding time information to places or transitions. In this way quantitative precedence relations can be stated by means of Petri nets.

The notion of time attached to the transitions as it has been presented 25 years ago by J.A. Meldman and A.W. Holt [MH71] was rather ambiguous. Was it a minimal delay, a maximal one, or an exact delay? Once a transition had been enabled, was it possible to cancel its firing or necessarily the transition was eventually fired when its attached duration elapsed? During these years, the intuitive notion of time has been refined into two notions: “timed” Petri nets and “time” Petri nets.

It has also been noticed that the time could be attached either to the transitions or to the places. Depending on the elements to which time is attached and on the fact that they were time or timed Petri nets, we get four representations with different expressiveness power which have been discussed in literature.

3.1 Attaching durations to places (p-timed)

When a simple duration is attached to the places, p-timed Petri nets are obtained. In ordinary Petri nets, transition firings are considered as instantaneous and the feeling was that associating a minimal sojourn time (delay) with the places was the simplest
formal way to introduce time [Sif77, Sif80]. This means that when a token enters a place \( p \), it is no longer available during the delay \( d(p) \) associated with this place. After this delay, the token turns available and can be taken into consideration in order to decide if the output transitions of place \( p \) are enabled and can be fired. It will be the case if the token load of their other input places are greater than one.

In such Petri nets, it is possible to model minimal delays but not maximal ones. Indeed it is impossible to fire a transition which is not enabled, but as in ordinary Petri nets, the firing of a transition can be delayed, for example to fire other enabled transitions.

Let us consider, for example, the net in figure 1: a token in the place “trial is on” represents a trial. The duration attached to the place corresponds to the trial duration. This duration starts when transition “trial begins”. At the end of the duration, transition “trial terminates” is enabled and fired. With such a model, it is impossible to state a maximal time constraint for the trial duration. In fact, if transition “trial terminates” has another input place denoting for example the receiving of a document from another court, if this document is never received (i.e. if the place never receives a token) then the trial will continue for ever because the transition “trial terminates” will never be enabled.

In addition there is an ambiguity: are transitions fired as soon as they are enabled? Actually, a transition frequently denotes the occurrence of an external event (the receiving of a letter for example) and the complete firing rule is: fire enabled transition synchronously with the occurrence of the attached event. As a consequence, a forward-looking analysis is not simple because only minimal delay are stated.

In a nutshell, the semantics involved in this model is that a place models an action (with a minimal duration).

### 3.2 Attaching a duration to transitions (t-timed)

As Petri nets comprise two types of nodes, it was natural to try to attach durations to transitions. In effect, it has been the first approach [MH71, Ram74] although it

Figure 1: p-timed Petri net
seems contradictory with the fact that transition firings were instantaneous actions in ordinary Petri nets.

In this approach, a minimal delay is associated with the transitions. This duration can be considered as a firing duration that expresses the duration of the represented action: transitions model actions. The enabling process for transitions is the same as the usual one but the firing process is changed: when firing a transition \( t \) with an associated delay \( d(t) \), the marking change requires two steps, each step being an instantaneous primitive. In the first one, tokens are deleted from the input places of \( t \) (or they are said to be reserved which means that they are no longer visible by other transitions) but tokens are not added to output ones. Then, a second step put tokens in output places, \( d(t) \) time units after the firing beginning.

Considering the same problem as in the previous case, we get the net on figure 2. As the previous one, this approach cannot model maximal time constraints. Actually a transition can remain enabled without being immediately fired, for example because its firing has to be synchronised with the occurrence of some event. This leads to an ambiguous firing time point if it is not assumed that any transition is fired as soon as it is enabled. In addition, some problem appears about the marking: tokens disappear during the firing. This means that the current marking has to be represented by two vectors, the first one denotes the tokens which are available for enabling transitions and the second one denotes the ones which are not visible because they are being reserved for the firing of some transition.

As it can be seen on this example, the nets on figure 2 and figure 1 are very similar ones and equivalence between the p-timed approach and the t-timed one has been stated [Sif80]. It is possible to translate any model into the other one. But the two main problems (no maximal delay and tokens not available) lead to the impossibility to represent interesting temporal mechanisms such as those called watch dogs or timeout in computer science. These mechanisms allow to simultaneously define a normal behaviour of the modelled system and what has to be done when some maximal time constraint is reached. The notion of timeout is very useful in many cases and we are going to see two other approaches based on time window attached either to places or transitions.
3.3 Attaching a time window to tokens in places (p-time)

In this approach [KDC96, Kha97] a time interval \([d_{\text{min}}(p), d_{\text{max}}(p)]\) is attached to each place \(p\). When a token is added in place \(p\), it is no longer available and remains so during \(d_{\text{min}}(p)\) time units. Between \(d_{\text{min}}(p)\) and \(d_{\text{max}}(p)\) the token is available. It can be used to verify if some output transition \(t\) of place \(p\) is enabled or not, and it can be used to fire \(t\) if it is enabled. After \(d_{\text{max}}(p)\) time units, the token is dead, this means that it can no longer be considered to enable or fire transitions.

In consequence, if no transition whose \(p\) is an input place becomes enabled and is 

fired during its availability interval the token becomes dead and is never more able to enable any transition. The example on figure 3 shows an example explicating an imprecise duration between close of pleadings and the trial beginning. If the event “preceding trial terminates” occurs too late, then the token in place \(pw\) dies and this 

denotes the fact that the constraints “the time interval between close of pleadings and the trial beginning cannot exceed 6 months” is violated.

Compared with the two previous approaches, the expressiveness is richer. It is 

possible to describe both maximal and minimal constraints. Let us note that, as in 

the p-time approach, the expressed constraints are only related to transitions which 

are output of place \(p\): they are independent from the environment. Considering the 

example, the interval between the pleading closure and the trial beginning does not 

depend of other actions. But a new problem arises: if the time constraints are not 

satisfied, dead tokens appear and the model is not able to describe alternative beha-

viours in such cases. The dead tokens only permit to exhibit the problem: what to do 

is not specified.

3.4 Attaching a time window to enabled transitions (t-time)

In the t-time approach, a time interval \([d_{\text{min}}(t), d_{\text{max}}(t)]\) is associated with each 

transition \(t\): it represents the enabling interval during which the transition can be fired [Mer76, Mer79].

The time counting begins when the transition is enabled and the firing can only 

occur during the defined time interval. This means that if the transition \(t\) is enabled 
at time \(\theta\), it can only be fired in the interval

\([\theta + d_{\text{min}}(t), \theta + d_{\text{max}}(t)]\)
So, the time interval bounds are respectively the shortest and the longest delay between the enabling time point and the firing one.

This model is a more general one than the p-timed or t-timed ones. It permits to represent minimal and maximal wait for events attached to transitions. Let us consider an example on figure 4: transitions \( tb \) and \( tt \) specify the trial beginning and termination.

An important point has to be noticed: between the enabling time point and the firing one, tokens are still available and can be used for firing another transition. It permits to describe pre-emption mechanism as shown on the example. If a new evidence is discovered while the trial is on, transition \( ne \) will be immediately fired because of its firing interval \([0, 0]\) and, although it is enabled, transition \( tt \) will not be fired.

Some particular cases can be described: when the firing interval is \([0, 0]\) it expresses an imperative immediate firing while when there are no timing constraints a time window \([0, \infty]\) is attached to the transition. A \([d, d]\) time interval denotes a precise firing that occurs exactly \(d\) time units after the transition enabling. Because of this descriptive power, this model is a very interesting one.

In addition, several methods have been developed to analyse the temporal behaviour of this model. In such nets the current marking is not sufficient to characterise the current state. It is necessary to build the list of enabled transitions and to attach to them the current firing intervals. Between two transition firings, all the states are, in some way equivalent, because they only differ by a translation of time. The states occurring just after transition firings can be taken as representants of the equivalent classes. The set of markings which will effectively be reached when temporal constraints are taken into account may be smaller than the set of reachable markings of the underlying ordinary Petri net because for some states, enabled transitions may not be effectively fired because of the temporal constraints.

Generating all the reachable classes allows a formal analysis of all the possible states of systems represented by t-time Petri nets. A general presentation of this approach can be found in [BD91]. Full details can be found in [Men82] and some
hints about a possible use for reasoning has been published in [TSA95].

A limitation of this formal analysis is the combinatorial explosion of the state classes and the imprecision of the characterisation of the duration of scenarios (some specific sequence of transition firings or set of sequences in the t-time Petri net) due to concurrent evolutions. An approach based on a linear logic representation of Petri nets is currently been developed to address the specific issue of the characterisation of scenarios durations [KPVC96]. This approach is able to give such a time interval by characterising some specific sequence or set of sequences in the net. The following section of this paper uses this model.

4 The civil litigation as an example

4.1 The problem description

We are interested in modelling procedures and time limits for an action between a plaintiff and a defendant. Once an action has been commenced by a plaintiff (and acknowledged) and a defence served by a defendant, rules and regulations, called “Automatic Directions”, take effect. This action can be divided into four main parts:

- the discovery of relevant documents between solicitors; it must be done within 14 days after close of pleadings. During this stage, each party has to inform the other one about documents in his possession which are connected to the case.

- the inspection of documents: 7 days are allowed for inspection by the other party.

- the exchange of expert evidence and non-expert witness statements must be done within 14 weeks after close of pleadings.

- the trial setting down: it consists in obtaining a trial date from the Court and must be done within 6 months after close of pleadings.

4.2 The t-time Petri net for the civil litigation action

The model in figure 5 is divided into two main parts: on the left side of the vertical grey line are described the plaintiff and defence behaviours while on the right part are described the automatic directions timing constraints.

The place and transition interpretations are:

- Places $pp_i$ and transitions $tp_i$ describe the plaintiff.

- Places $pd_i$ and transitions $td_i$ describe the defendant.

- Transition $tdl$ denotes the delivery of defence while $tp1$ denotes the plaintiff beginning of the action.
Figure 5: Civil litigation model
• Places \( p_2, p_3, p_4, p_5 \) denote the exchange documents process. Transitions \( t_1, t_2 \) describe communications between the parties.

• Transitions \( t_3, t_4, t_5, t_6 \) describe common actions of the two solicitors (synchronisation points at the court when time constraints are satisfied). They respectively denote the end of discovery stage, the end of inspection, the end of exchange stage and the trial setting down.

• Places \( p_l \) and transitions \( t_l \) represent the time constraints in automatic directions. For example, time constraints associated with \( t11, t13 \) and \( t14 \) are relative to the “close of pleadings” and transition \( t12 \) is relative to the end of discovery stage (transition \( t3 \)).

The time windows associated to the transitions are the following ones:

\[
\begin{align*}
    t_3 : & [0\text{days}, 14\text{days}] & (1) \\
    t11 : & [14\text{days}, 14\text{days}] & (2) \\
    t4 : & [0\text{days}, 7\text{days}] & (3) \\
    t12 : & [7\text{days}, 7\text{days}] & (4) \\
    t5 : & [0\text{weeks}, 14\text{weeks}] & (5) \\
    t13 : & [14\text{weeks}, 14\text{weeks}] & (6) \\
    t6 : & [0\text{months}, 6\text{months}] & (7) \\
    t14 : & [6\text{months}, 6\text{months}] & (8)
\end{align*}
\]

Other ones have time windows corresponding to an evaluation of the actual duration of the associated activities. For example, the duration \([d_{\min}(t1), d_{\max}(t1)]\) corresponds to the approximate duration between the time point at which the documents are posted by the plaintiff and their delivery to the defendant by the mailing service.

### 4.3 Some temporal properties

Transitions \( t_l \) are fired when time limits are reached because they have a precise time interval attached to them. They let some pending tokens in the Petri net showing the state of the litigation when the constraint violation occurred. It is an important information for the litigation continuation because it allows to quickly detect which constraint is violated. For example, if inspection of documents do not occur within 7 days after the discovery stage, transition \( t12 \) fires and tokens in places \( pp5 \) and \( pd5 \) will be pending ones because transition \( t4 \) is no more fireable.

Because nothing is specified about the future evolution after such constraint violation, the transitions \( t_l \) have no output places but any behaviour could be represented as output places of these transitions.

An example of temporal constraint consistency analysis could be the verification that the 14 days maximal delay for the discovery stage is consistent with the durations of the procedures that the plaintiff and the defendant have to execute (gather documents and send them). From the structure of the Petri net, it can be formally derived that the maximal delay in order to put a token in place \( pp3 \) is equal to the
maximum value of the maximal length of the paths \((tp1,tp2)\) and \((td1,t2,tp2)\). Symmetrically, the maximal delay to put a token in place \(pdt3\) is the maximum value of the maximal length of the paths \((tp1,t1,td2)\) and \((td1,td2)\). The guarantee that no constraint violation will occur (transition \(t3\) can fire before the time limit, represented by \(tl1\), expires), will only be proved if the maximum of these two maximal delays is less than 14 days.

An interesting point is that 4 paths have been involved in the reasoning. Two of them are evident \((tp1,tp2)\) and \((td1,td2)\) because they simply correspond to the activity of the plaintiff and of the defendant during the corresponding phase. The two other ones \((tp1,t1,td2)\) and \((td1,t2,tp2)\) are more subtle because they comprise actions of the plaintiff and of the defendant (and also of the mailing service). These paths and the associated temporal intervals can be constructed using linear logic characterization [KPVC96].

5 Conclusion

The example presented here was intended to illustrate the descriptive power of some extension of Petri nets taking into account explicit, qualitative, temporal constraints. Their adequacy to represent and analyse the temporal structures of legal procedures have been pointed out.

It is clear that a lot of work has to be done in order to enrich this approach and show that actually most of the temporal constraints of this application domain can be represented and analysed.

Petri nets are currently been used in many applications, mainly in computer science; tools have been developed which could be adapted to the domain of legal systems. Another important point is that, recently, Petri nets have turn to be the object of many studies in the field of artificial intelligence and formal logic because the limitation of classical logic for reasoning about dynamic systems is now recognised. Probably, some breakthrough will occur in this field in the next years and this will enrich the possibilities of applying Petri nets for the analysis of temporal structures of legal procedures.

Acknowledgments:

We are very grateful to E. Nissan and D.Y. Farook for expliciting us the procedure of a civil litigation action and its associated Automatic Directions.

References


