Abstract. We present an approach to construct the occurrence graph for ITCPN (Interval Timed Coloured Petri Nets). These models, defined by Van Der Aalst in [VAN] can simulate other timed Petri nets and allow to describe large and complex real-time systems. We define classes as sets of states between two occurrences, and we use these classes to define the occurrence graph of an ITCPN. Then an equivalence relation based on time is defined for classes, and we show that occurrence graphs reduced using this equivalence relation are finite if and only if the set of reachable markings is finite. These graphs can be used to verify all the dynamic properties such as reachability, boundedness, home, liveness and fairness properties but also performance properties: minimal and maximal bounds along a occurrence sequence or a cycle. Finally we complete delay based equivalence with a colour based equivalence in order to achieve further reduction.

Keywords: interval timed coloured Petri nets, occurrence graph.

1 - Introduction

As formal verification techniques are wide spreading the interest for timed models increases constantly [SIF91], either in real times area with synchronous language like ESTEREL ([BERR]) or the protocol area with various timed LOTOS [QUE]. This seems quite normal since, in the common opinion, a model is correctly functioning not only if it has good behavioural properties such liveness and fairness but also if he has good time performances. Interest for time in Petri nets is quite old (Ramchandany paper [RAM] appears in 1973), but until now the various proposed models are not considered as totally adequate to model real systems. First timed Petri nets of [RAM], associated deterministic time to transitions or places [SIF,77] and were applicable only to few production systems or electronic devices where the time where for performing an operation is constant. The behavioural properties of these models can be investigated only if they allow finite firing sequences or if there is a steady state when infinite firing sequences are allowed. To allow modelling of a larger set of systems, Florin and Natkin have defined Stochastic Petri nets [FLO] where stochastic delays are associated with transitions. If these delays are distributed according to negative exponential probability laws, it is possible to translate the net into a continuous Markov chain. This Markov chain may then be analysed to gives probability laws of various quantities like mean firing period, mean residence time in a given state and so on. Besides these various performance measures, this model allows also to investigate usual behavioural properties. Despite its obvious interest, this model is still unsatisfactory for systems where strict time limits must be respected, as real time systems or protocols. By the way, negative exponential probability distribution allows to fire a transition after a delay ranging from 0 to infinity. So it is impossible to prove, for instance that a timer will send a signal before a given delay or not. Conversely a maximum firing delay for a transition may change the behavioural properties of the untimed underlying net, changing an unbounded but subject to deadlock net into a bounded and live one, since liveness is not monotonic with marking. So a model with time constraints defined by intervals is needed, and it must be analysed without removing time constraints. Such a model has been defined by Merlin [MER] which has proposed to associate an interval, specifying the minimal and maximal firing delays, with each transition. Berthomieu and Menasche in [BERT,83] have defined a procedure to obtain an occurrence graph for this model but this procedure is slow since it implies at each step (i.e. each firing), the resolution of a system of inequations. Moreover, Merlin's model is "uncoloured" and is difficult to use for modelling of complex systems.

Van der Aalst ([VAN]) recently proposed Interval timed coloured Petri Nets, also called ITCPN, which are Coloured Petri Net (CPN) extended with time. A time stamp is associated with each token, specifying
the minimal time from which this token could be consumed. When a token is created, its time stamp is
defined by adding to the firing time of the creating transition, a value chosen randomly in an interval
depending on three parameters: colours of consumed tokens, colour of the created token and arc used.

ITCPN are extremely general since they can simulate other existing models, deterministic one's with
intervals reduced to one value, as well as Merlin model with intervals depending only on the transition.
They can also simulate Stochastic Petri nets since bounds of intervals may range from 0 to \( \infty \), and one
could specify a probability distribution for each interval. Consequently they allow to describe large and
complex real-time systems.

The same author has proposed an analysis method, called the Modified Transition System Reduction
Technique (MTSRT), which is "sound" i.e. any occurrence sequence in the initial model is possible also
in the modified one, but not "complete", i.e. the converse is not true (some occurrence sequence in the
modified model does not reflect any occurrence sequence of the initial model). Moreover, for nets allowing
infinite occurrence sequences, MTSRT would lead to infinite graphs.

We present here an approach to built ITCPN occurrence graphs which have not these drawbacks.
Firstly, we give basic definitions on ITCPN and ITCPN behaviour in section 2. In the following section
we point out a simple condition for the occurrence of an event after \( n \) event occurrences. Then we define
classes as sets of states between two occurrences and we define the occurrence graph of an ITCPN. In
section 4 we define an equivalence relation based on delays, and we show that an occurrence graph, using
this equivalence relation to fuse classes, is finite if and only if the set of reachable markings is finite (i.e.
even if the net allows infinite occurrence sequences). It can be used to verify all the dynamic properties
such as reachability, boundedness, home, liveness and fairness properties. Fifth section is devoted to the
calculus of minimal and maximal delays for an occurrence sequence, cycling or not. In section 6, we
show how to complete delay based equivalence with a colour based equivalence in order to achieve further
reductions of an occurrence graph. We end with an example in section 7.

2 - Interval Timed Coloured Petri Nets

We will introduce here only necessary definitions and notations. For further details we refer to [JENa] for
coloured Petri nets and to [VAN] for Interval timed coloured Petri nets. Let's just recall that \( \mathbb{N} = \{\ldots, 0, 1, 2, 3, \ldots\} \)

An interval timed coloured Petri net is a coloured net where time stamps are attached to tokens when they
are created. Time stamps must be chosen inside an interval associated with the arc creating each token.

**Definition 1**

\( TS \) is the time set, \( TS = \{x \in \mathbb{R} : 0 \leq x \} \), i.e. the set all non-negative reals. \( \text{INT} = \{[y, z] \in \text{TS} \times \text{TS} : 0 \leq y \leq z \} \) represents the set of all closed intervals. If \( x \in \text{TS} \) and \( [y, z] \in \text{INT} \), then \( x \in [y, z] \) if and only if \( y \leq x \leq z \).

An interval timed coloured Petri net is defined as follows:

**Definition 2**

An Interval Timed Coloured Petri Net is a five tuple \( \text{ITCPN} = (\Sigma, P, \text{T}, C, F) \) where

(i) \( \Sigma \) is a finite set of types, called colour sets.
(ii) \( P \) is a finite set of places.
(iii) \( \text{T} \) is a finite set of transitions.
(iv) \( C \subseteq P \rightarrow \Sigma \), \( C(p) \) is the type of \( p \) which specifies the set of allowed values (or colours) for tokens of
place \( p \).
(v) \( \text{CT} = \{(p, v) : p \in P \text{ and } v \in C(p)\} \) is the set of all possible coloured tokens
(vi) \( F \) is the transition function: \( F(t) \) specifies which tokens are consumed and produced by firing
transition \( t \) and also the interval in which their time stamps must be chosen.

\( F(t) \in \text{CT}_{\text{MS}} \rightarrow (\text{CT} \times \text{INT})_{\text{MS}} \)