A specification of a hybrid Petri net semantics for the HISim simulator.

Alberto Amengual

TR-09-003

March, 2009

Abstract

Affiliations: International Computer Science Institute & Universitat de les Illes Balears

In this paper we formally define the hybrid Petri net structure and evolution semantics implemented in the HISim simulator,[1] which was developed as part of my thesis. The semantics described here is based on an HPN formalism called differential Petri nets (DPN; Demongodin and Koussoulas, 1998).

The main reasons why we chose DPNs among the multiple available HPN formalisms are that, on the one hand, it provides great flexibility and on the other, that it incorporates elements that can help dealing with the potential accuracy issues that affect other formalisms. Among the features that provide more flexibility than other HPN semantics are the fact that the marking of continuous places (called differential places in DPN) can take any value in $\mathbb{R}$, either positive and negative, and also that the maximal firing speed of continuous transitions (called differential transitions) may be a linear or nonlinear function of the markings of the input continuous places connected to it. With respect to mechanisms for dealing with potential accuracy issues, every continuous transition is associated with an individual delay representing its step of integration, therefore a shorter step can be associated with continuous transitions that require a more accurate approximation without penalizing other transitions or increasing the overall simulation load.

a. Two new types of arcs are introduced, inhibitory and test arcs. Semantics for these types of arcs will be described shortly, but in the mean time we can describe them as follows; both test and inhibitory arcs are used in relation to the enabling of transitions, but not to their firing, and in line with that, they can only be used to link an input place to a transition but not a transition to an output place. Test arcs are similar to resource arcs in their role in the enabling of transitions, but they don't consume any tokens (or continuous marking) upon firing of the transition. Inhibitory arcs work by preventing a transition to fire if the input place's marking is at least equal to the weight of the arc, and they don't consume any tokens either.

[1]HISim is open source software and can be downloaded from the Sourceforge web site, at http://sourceforge.net/projects/hisim.

b. Links between a discrete place and a continuous transition are not allowed, even if a reciprocal arc with the same weight exists in the other direction. There is no overall loss in expressive power with respect to DPNs, as the same effect is obtained connecting a discrete place to a continuous transition using a test arc, which is allowed. Nevertheless, there are some differences between the two configurations, e.g. a test arc does not generate conflicts, while a normal (resource) arc can. Another difference is that test arcs don't reserve tokens, therefore an enabled transition can become disenabled during its delay time as a result of a change in a condition on a test (or inhibitory) arc.

c. There is no marking decomposition for continuous places, and a race firing policy is established for discrete transitions. As we will shortly see, DPNs decompose the marking, both discrete and continuous, in reserved on the one hand, and available or non-reserved on the other. In our view this can create inconsistencies in cases where a continuous place is linked with a discrete transition, and consequently in the semantics described here, discrete marking will be decomposed but not continuous marking. In a closely related issue, DPNs use a pre-selection policy for discrete transitions, i.e. when they become enabled they reserve the marking needed to fire therefore guaranteeing that they will be able to fire after their delay time. This no longer holds in the semantics described here, because there is no decomposition of the marking of continuous places (and consequently it can't be reserved), and also because test and inhibitory arcs don't reserve marking either. As a result, a discrete transition can become disenabled during its delay time.

d. Capacity for discrete places. A finite capacity can be associated to a discrete place, which will establish an upper limit on its marking. Capacity can also be defined as infinite, in which case obviously there will be no limit on the place's marking.

Let's now define the formal semantics. The discussion is organized following the structure in (Demongodin and Koussoulas, 1998) in order to facilitate comparison between the two specifications.

HPN Structure

Definition 1. A Hybrid Petri net is defined by $B = \{R, f_{nt}, f_{at}, m_0, \tau, f_c\}$ verifying the following conditions.

1. R is a Petri net defined by $\{P, T, A, Pre, Post\}$ with
   - P finite set of places with $|P| = n_P < \infty$.
   - T finite set of transitions with $|T| = m_T < \infty$.
   - $A \subseteq (P \times T) \cup (T \times P)$: finite set of directed arcs.
   - $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.
   - $Pre : P \times T \rightarrow \mathbb{R}$ or $\mathbb{N}_+$ is a function that defines the weight of arcs from a place to a transition.
   - $Post : P \times T \rightarrow \mathbb{R}$ or $\mathbb{N}_+$ is a function that defines the weight of arcs from a transition to a place.
2. $f_{nt} : P \cup T \rightarrow \{D, C\}$, named "node type function", indicates for every node whether it is a discrete or a continuous node.
3. $f_{at} : A \rightarrow \{R, I, E\}$, named "arc type function", indicates for every arc whether it is a resource, inhibitory, or test arc.
4. $m_0 : P \rightarrow \mathbb{R}$ or $\mathbb{N}$ is the initial marking.
5. τ is a map, to be called timing map, that associates a delay (a non-negative real number) to every transition (for continuous transitions it represents the step of integration) and additionally a maximal firing speed for each continuous transition.
6. $f_c : P \rightarrow \mathbb{N}_+ \cup \{\infty\}$, named capacity function, associates to every place a finite or infinite capacity. Additionally it is imposed that $f_c(P_i) = \infty$ for every continuous place, i.e. $f_{nt}(P_i) = C$.

Regarding Pre and $Post$ functions:

- For a discrete place, i.e. if $f_{nt}(P_i) = D$, then $Pre(P_i, T_j)$ and $Post(P_i, T_j)$ are positive integers.

- For a continuous place, i.e. if $f_{nt}(P_i) = C$, then $Pre(P_i, T_j)$ and $Post(P_i, T_j)$ are real numbers.

Denoting by $m_i(t)$ the marking of a place P_i at time t, then its value depends on the type of place:

- For a discrete place, the marking is a non-negative integer, i.e. if $f_{nt}(P_i) = D$, then $m_i(t) \in \mathbb{N}$. If the place has finite capacity, i.e. $f_c(P_i) = c \neq \infty$, then $m_i(t) \leq c$.

- For a continuous place, the marking is a real number, i.e. if $f_{nt}(P_i) = C$, then $m_i(t) \in \mathbb{R}$. If the place has finite capacity, i.e. $f_c(P_i) = c \neq \infty$, then $m_i(t) \leq c$.

Negative or non-integer number weights are not allowed in arcs connected to discrete places as it may result in a negative or non-integer marking, which is not allowed.

Regarding the timing map τ:

- For a discrete transition, i.e. if $f_{nt}(T_j) = D$, then $\tau(T_j) = d_j$ is the delay associated to the transition T_j, expressed in time units.

- For a continuous transition, i.e. if $f_{nt}(T_j) = C$, then $\tau(T_j) = \langle V(T_j)|d_j\rangle = \langle Vj|h\rangle$, where V_j is the maximal firing speed associated with the transition T_j, expressed in parts per time unit, and h is the time constant that will be used as the step in the integration of the differential equation. The maximal firing speed may be a constant, a linear combination, or a nonlinear function of the markings of the continuous places connected to the continuous transition. As we will shortly see, we will define a condition so that inhibitory arcs are not allowed when connecting a continuous places to a continuous transition. Therefore we have:

$$V_j = c \text{ or } V_j = \sum_{i=1}^{n} a_i m_i, \text{ or}$$
$$V_j = g(m_1, m_2, ..., m_n), \text{ where } m_i = m(P_i),\ f_{nt}(P_i) = C$$

The maximal speed cannot be a function of derivatives or integrals of the markings since only the instantaneous values are available at a given time. As the markings evolve in time, the maximal speed will change. At a given point in time, besides the maximal speed V_j, an instantaneous speed v_j is associated with every continuous transition, which can be either zero or the maximal speed:

$$v_j(t) = V_j, \text{ or } v_j(t) = 0$$

The authors point out that this schema can represent in principle any form of discrete approximation of an ordinary differential equation besides the simple Euler approximation used in the paper, and in particular they refer to variable-step methods, where some type of logic is used to decide on the magnitude of each step.

Conditions on the HPN Structure

Condition 1. No resource arcs are allowed linking a discrete place to a continuous transition (in both directions). This conditions is established in order to guarantee that the marking on a discrete place is always a non-negative integer. In DPNs, in order to guarantee the non-negative, integer marking of discrete places, a different condition is established, namely that resource arcs linking a discrete place and a continuous transition are allowed only if for every such arc, a reciprocal one exists that has the same weight, therefore the firing of the continuous transition will have no net effect on the discrete place's marking.

A link between a discrete place and a continuous transition is a fundamental part of hybrid semantics: on the one hand, it is used as the main mechanism for discretizing the continuous transitions (e.g. Demongodin and Koussoulas, 1998, p. 574); on the other hand it allows discrete control of continuous processes. In the semantics described here, test arcs linking discrete places and continuous transitions are used to realize these important functions, the equivalent to a pair of reciprocal arcs of equal weight being a single test arc of the same weight from the discrete place to the continuous transition. The semantics of test and inhibitory arcs will be described later in the document, where the evolution rules are introduced.

Condition 2. No inhibitory arcs are allowed linking a continuous place to a continuous transition. A continuous transition is always enabled, and the speed of the continuous process it models is controlled through its firing speed, that as we have seen can be a function of the marking of the continuous places that are linked to it. On the other hand, the continuous process can be abruptly interrupted as a result of its links to discrete places (see condition 1). Therefore, in order to keep the semantics coherent, this type of abrupt interruption is not allowed to be performed using a continuous place.

Condition 3. No test or inhibitory arcs are allowed from any type of transition to any type of place. As we will see in the next section, the semantics of test and inhibitory arcs has to do with the enabling of transitions and not with their firing, therefore it makes no sense to have this type of links going in the direction from a transition to a place.

Marking of the HPN

In DPNs, the marking of the HPN is decomposed in reserved and non-reserved. At any time, the marking m is the sum of the reserved marking m^r and the non-reserved marking m^n. If we denote by $m_i(t)$ the marking of a place P_i at time t, then we have $m_i(t) = m_i^r(t) + m_i^n(t)$. The authors describe the dynamics under which marking is reserved as follows: when a discrete transition becomes enabled at time t, the tokens (or continuous marking, in case of an input continuous place) required to fire this transition are reserved during the associated delay. When the delay is over, the transition is fired and the tokens (or continuous marking) reserved for firing are removed from the input places of this discrete transition, while non-reserved tokens (or continuous marking) are added to the output places. Continuous transitions don't reserve marking.

The decomposition of the marking defined in this way, though, can lead in our view to a somewhat incoherent semantics when applied to continuous places. Consequently, we decided not to decompose the marking of continuous places, which corresponds to change (c) introduced at the beginning of this document. As it was mention there, this change and the fact that discrete places connected via test or inhibitory arcs to a discrete transition are not subject to pre-selection implies a race condition on discrete transitions.

Evolution Rules

The state of an HPN is defined by the marking of its places, which evolves as a result of the firing of transitions. In this section we introduce the definitions relative to the enabling and firing of discrete and continuous transitions. The following notation will be used: T_j^-, T_j^+ are the sets of input and output places, respectively, of the transition T_j.

Definition 2. A discrete transition, $f_{nt}(T_j) = D$, is enabled at time t iff: Each discrete input place linked to the transition via a resource arc has a non-reserved marking at least equal to the weight of the arc:

$$\forall P_i \in T_j^-; f_{nt}(P_i) = D, (P_i, T_j) \in A; f_{at}(P_i, T_j) = R \Rightarrow m_i^n(t) \geq Pre(P_i, T_j).$$

AND Each continuous input place linked to the transition via a resource arc has a marking at least equal to the weight of the arc:

$$\forall P_i \in T_j^-; f_{nt}(P_i) = C, (P_i, T_j) \in A; f_{at}(P_i, T_j) = R \Rightarrow m_i(t) \geq Pre(P_i, T_j).$$

AND Each input place linked to the transition via a test arc has a marking at least equal to the weight of the arc:

$$\forall P_i \in T_j^-, (P_i, T_j) \in A; f_{at}(P_i, T_j) = E \Rightarrow m_i(t) \geq Pre(P_i, T_j).$$

AND Each input place linked to the transition via an inhibitory arc has a marking less than the weight of the arc, in other words the input place inhibits the transition if it has a marking at least equal to the weight of the arc:

$$\forall P_i \in T_j^-, \forall (P_i, T_j) \in A; f_{at}(P_i, T_j) = I \Rightarrow m_i(t) < Pre(P_i, T_j).$$

Note that only the first condition deals with non-reserved tokesn. For all other cases the condition is defined on the marking, whether it is reserved or not. Note also that there is no q-enabling of discrete transitions as defined for example in (David and Alla, 2005), or in other words discrete transitions can only be 1-enabled.

As it was mentioned previously, when a discrete transition becomes enabled tokens are reserved in each input discrete place linked with the transition via a resource arc, the amount to be reserved being equal to the weight of the correspond arc. No tokens are reserved either in places linked to the transition via test or inhibitory arcs or for continuous places.

Definition 3. A continuous transition, $f_{nt}(T_j) = C$, is enabled at time t iff: Each discrete input place linked to the transition via a test arc has a marking at least equal to the weight of the arc:

$$\forall P_i \in T_j^-; f_{nt}(P_i) = D, (P_i, T_j) \in A; f_{at}(P_i, T_j) = E \Rightarrow m_i(t) \geq Pre(P_i, T_j).$$

AND Each discrete input place linked to the transition via an inhibitory arc has a marking less than the weight of the arc:

$$\forall P_i \in T_j^-; f_{nt}(P_i) = D, (P_i, T_j) \in A; f_{at}(P_i, T_j) = I \Rightarrow m_i(t) < Pre(P_i, T_j).$$

Note that a continuous transition is subject to enabling conditions only as a result of its links to discrete elements, i.e. under the hybrid semantics, but as long as it is only linked to continuous places, it would be continuously enabled. Note also that discrete input places are not allowed to be linked to a continuous transition via a resource arc (see change (b) at the beginning of the document) and that input continuous places are not allowed to be linked to a continuous transition via an inhibitory arc.

Continuous transitions' enabling do not produce token reservation.

Let's now focus on the firing rules.

Definition 4. An enabled discrete transition will fire at the end of its delay interval (given by the timing map τ), which starts the moment the transition becomes enabled. When a discrete transition is fired:

- For every discrete input place linked to the transition via a resource arc, a number of reserved tokens equal to the weight of the arc are removed from the place.

- For every continuous input place linked to the transition via a resource arc, an amount of marking equal to the weight of the arc is removed from the place. For a continuous place, the arc weight can be negative, and if this is the case, then actually a negative amount of marking will be removed from the place, i.e. the marking of the place will increase.

- For every discrete output place, an amount of non-reserved tokens equal to the weight of the arc is added to the place, if that can be done without exceeding the place's capacity, $f_c(P_i)$. Otherwise non-reserved tokens will be added only until the capacity is reached, i.e. $f_c(P_i) - m_i(t)$ tokens. A discrete transition's ability to fire is not affected by its output place's capacity having been reached.

- For every continuous output place, an amount of tokens or marking equal to the weight of the arc is added to the place. If the arc weight is negative, a negative amount of marking will be added in the output place, resulting in an actual decrease in the place's marking.

$f_{nt}(T_j) = D,$ and $\tau(T_j) = d_j, T_j$ is fired at time $t + d_j \Rightarrow$

$\forall P_i \in T_j^-; f_{nt}(P_i) = D, (P_i, T_j) \in A; f_{at}(P_i, T_j) = R \Rightarrow$

$m_i^r(t + d_j) = m_i^r(t) - Pre(P_i, T_j)$

$\forall P_i \in T_j^-; f_{nt}(P_i) = C, (P_i, T_j) \in A; f_{at}(P_i, T_j) = R \Rightarrow$

$m_i(t + d_j) = m_i(t) - Pre(P_i, T_j)$

$\forall P_i \in T_j^+; f_{nt}(P_i) = D, (P_i, T_j) \in A \Rightarrow$

if $m_i(t) + Post(P_i, T_j) > f_c(P_i)$ then $m_i^n(t + d_j) = m_i^n(t) + (f_c(P_i) - m_i(t))$

else $m_i^n(t + d_j) = m_i^n(t) + Post(P_i, T_j)$

$\forall P_i \in T_j^+; f_{nt}(P_i) = C, (P_i, T_j) \in A \Rightarrow m_i(t + d_j) = m_i(t) + Post(P_i, T_j)$

Note that a transition's delay can be zero, in which case the transition will fire immediately after becoming enabled. Note also that a discrete transition can become disenabled during the delay interval as a result of a race condition on an input place linked to it via a test or inhibitory arc, or a continuous input place. In that case the next time the transition becomes enabled, the delay will start over.

Next we are going to deal with the firing rule for continuous transitions. Before we actually introduce the final definition let's start with an approximation.

Pre-Definition 5. An enabled continuous transition will fire. When a continuous transition is fired:

- For every (continuous) input place linked to the transition via a resource arc, an amount of marking equal to the weight of the arc multiplied by the firing speed is removed from the place.

- For every (continuous) output place, an amount of marking equal to the weight of the arc multiplied by the firing speed is added to the place.

Because we are dealing with a continuous transition, we assume it will fire for a certain interval of time *dt*. The amounts of marking removed or added to the places linked to a continuous transition depend on the amount of time during which the transition has fired, therefore we obtain:

$$
\begin{aligned}
&f_{nt}(T_j) = C, \text{ and } v_j(t) \text{ firing speed of } T_j \text{ at time } t; \\
&T_j \text{ is fired at time } t \text{ during an interval } dt \Rightarrow \\
&\forall P_i \in T_j^-, (P_i, T_j) \in A; f_{at}(P_i, T_j) = R \Rightarrow \\
&\quad m_i(t + dt) = m_i(t) - v_j(t) \cdot Pre(P_i, T_j) \cdot dt \\
&\forall P_i \in T_j^+, (P_i, T_j) \in A \Rightarrow \\
&\quad m_i(t + dt) = m_i(t) + v_j(t) \cdot Post(P_i, T_j) \cdot dt
\end{aligned}
$$

The firing speed of T_j at time $t, v_j(t)$ is called the *instantaneous speed* at time t, and following (Demongodin and Koussoulas, 1998) it is computed this way: if a continuous transition T_j is not enabled at time t, then $v_j(t) = 0$; if it is enabled, then $v_j(t) = V(T_j)$, i.e. the maximal firing speed associated with the transition by the timing map τ.

Now we need to deal with the firing interval *dt*. In order to do that, we need to understand the mechanism used to discretize continuous transitions, which is at the core of the particular hybrid semantics used here, i.e. representing continuous elements in a "Discrete-Event World". The technique consists in associating to every continuous transition an implicit discrete transition, linking the two of them by a discrete place in the following configuration:

The dynamics of this configuration are based on the fact that the discrete transition uses a pre-selection policy, i.e. reserves tokens it needs for its next firing, while the continuous transition doesn't. There is only one token in the discrete place, which is unavailable to the continuous transition most of the time, as the discrete transition keeps it reserved during its delay times d_j. Only when the discrete transition fires, the token is consumed and added back to the discrete place, and the continuous transition can use it to get

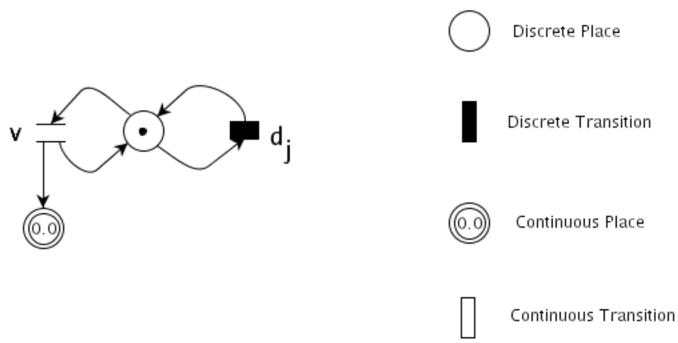

Figure 1: *Implicit discrete transition associated to a continuous transition.*

enabled until the discrete transition reserves it again. Because the firing time of the discrete transition is immediate, i.e infinitesimally small, the amount of time during which the token is available to the continuous transition is also infinitesimally small.

Is summary, the continuous transition is “forced” to fire only for an infinitesimally small amount of time every d_j units of time, i.e. it is discretized, adopting the implicit discrete transitions' delay d_j as its own.

Definition 5. An enabled continuous transition will fire at the end of its delay interval (given by the timing map τ), which starts the moment the transition becomes enabled. When a continuous transition is fired:

1. For every (continuous) input place linked to the transition via a resource arc, an amount of marking equal to the weight of the arc multiplied by the immediate firing speed multiplied by the delay interval is removed from the place.

2. For every (continuous) input place linked to the transition via a resource arc, an amount of marking equal to the weight of the arc multiplied by the immediate firing speed multiplied by the delay interval is removed from the place.

3. For every (continuous) output place, an amount of marking equal to the

weight of the arc multiplied by the immediate firing speed multiplied by the delay interval is added to the place.

$f_{nt}(T_j) = C, \tau(T_j) = \langle V(T_j)|d_j\rangle$, and $v_j(t) = V(T_j)$ is the immediate firing speed of T_j at time t (because T_j is enabled at time t, its immediate firing speed is made equal to its maximal speed given by the timing map);

$$\begin{aligned}
&T_j \text{ is fired at time } t + d_j\\
&\forall P_i \in T_j^-, (P_i, T_j) \in A; f_{at}(P_i, T_j) = R \Rightarrow\\
&\quad m_i(t + d_j) = m_i(t) - V(T_j)\cdot Pre(P_i, T_j)\cdot d_j\\
&\forall P_i \in T_j^+, (P_i, T_j) \in A \Rightarrow\\
&\quad m_i(t + d_j) = m_i(t) + V(T_j)\cdot Post(P_i, T_j)\cdot d_j
\end{aligned}$$

If we compare the consumption and production expressions in definition 5 with the temporary definition introduced previously, it will be noted that the firing interval of the continuous transition dt has been substituted in the final version by the firing delay d_j. That means that, after waiting for d_j units of time, the continuous transition fires for an infinitesimally small amount of time (as discrete transitions do) but the amounts of marking consumed and produced correspond to those that would have been produced and consumed had the transition fired for d_j units of time. Or in other words, the transition has "virtually" fired for d_j units of time, the difference (with respect to a continuous firing) being that the speed $v_j(t) = V(T_j)$ has been kept constant as a result of the discretization (remember that the maximal speed $V(T_j)$ can be a function of the marking of the continuous input places to the transition). Consequently, the smaller the delay dj, the more frequently $v_j(t)$ will be evaluated and the better the discrete approximation will be.

References

David, R. and Alla, H. (2005). *Discrete, Continuous, and Hybrid Petri Nets.* Springer-Verlag, Berlin, Heidelberg.

Demongodin, I. and Koussoulas, N. (1998). Differential petri nets: Representing continuous systems in a discrete-event world. *IEEE Transactions on Automatic Control*, 43(4):573–579.