DYNAMIC REWRITING DIFFERENTIAL PETRI NETS FOR DISCRETE-CONTINUOUS MODELLING OF COMPUTER SYSTEMS

E. Guţuleac
Technical University of Moldova

INTRODUCTION

Performance modelling is concerned with the description, analysis and optimization of the dynamic behavior of computer systems. Several methods for verification and performance modeling of computer systems are based on different formal models [2]. Among the formalisms that are used, Petri nets (*PN*) are the most popular. A number of different classes of *PN* have been proposed [2, 8]. Timed Hybrid Petri nets (*HPN*) are *PN* based models in which some places may hold a discrete number of tokens and places a continuous quantity represented by a real number [1, 3].

To make design issues and analysis procedures more transparent, we tried to deviate as little as possible from the concepts and tenets of *HPN*. Thus, we created our extension building on relevant previous works: *HPN*, Generalized Stochastic Petri Nets [2] and Differential Petri Net (*DPN*) [4]. The result is a new type of *HPN*, which we call *Generalized DPN* (*GDPN*), and is able to represent the behaviour of continuous systems and discrete systems in a common model. The novel features of *GDPN* are accepted the negative-continuous place capacity, negative real values for continuous place marking and negative token-dependent arc cardinalities, that permit to generalize the concept of *HPN* and *DPN*. The *GDPN* is formed by three kinds of discrete and continuous objects: places, transitions and arcs. Places represent some kind of resources, transitions denote actions or events that happen in the system and arcs link the first two kinds of objects together, implementing the logic of the *GDPN*; they assign actions to resources, and vice versa.

To our knowledge existing methods do not support marked-controlled reconfiguration of systems. The purpose of this paper is to introduce *descriptive self-rewriting GDPN* that can dynamically modify their own structures by rewriting rules transitions some of their components thus supporting structural dynamic changes within modeled systems.

1. LABELED GENERALIZED DPN

Various extensions have been made to the framework of timed *HPN* since its in [ ]. In this section, we present a variant extended HPN, called labeled *GDPN*, which is is derived with customary notation from [3, 4]. Let L be a set of labels $L = L_P \cup L_T$, $L_P \cap L_T = \varnothing$. Each place p_i labeled $l(p_i) \in L_P$ has a local state and transition t_j has action labeled as $l(t_j) \in L_T$.

Definition 1: A labeled *GDPN* is a 11-tuple $H\Gamma$ =<*P*, *T*, *Pre*, *Post*, *Test*, *Inh*, K_p, K_b, *G*, *Pri*, *l* >, where: *P* is the finite set of places partitioned into a set of discrete places P_D, and a set of continuous places P_C, $P=P_D \cup P_C$, $P_D \cap P_C = \varnothing$. The discrete places may contain a natural number of tokens, while the marking of a continuous place is a real number (fluid level). In the graphical representation, a discrete place is drawn as a single circle while a continuous place is drawn with two concentric circles;

- *T* is a finite set of transitions, that can be partitioned into a set T_D of discrete transitions and a set T_C of continuous transitions, $T = T_D \cup T_C$, $T_D \cap T_C = \varnothing$. A discrete transition $t_j \in T_D$ is drawn as a black bar and continuous transition $t_i \in T_C$ is drawn as a rectangle;
- *Pre*, *Test* and $Inh: P \times T \to Bag(P)$ respectively is a forward flow, test and inhibition functions. $Bag(P)$ is a discrete or continuous multiset over *P*. The backward flow function in the multisets of *P* is a $Post: T \times P \to Bag(P)$, where define the set of arcs *A* and it describes the marking-dependent cardinality of arcs connecting transitions with places and vice-versa. The *A* is partitioned into subsets: A_d, A_s, A_h, A_c and A_t. The subset A_d and A_s contains the *discrete normal* arcs and *continuous set* of arcs which can be seen as a function: A_d:$((P_D \times T_D) \cup (T_D \times P_D)) \times IN_+^{|P|} \to IZ$ and the A_s:$((P_C \times T_D) \cup \{T_D \times P_C)) \times IR^{|P|} \to IR$, respecively . The arcs of A_d and A_s, are drawn as single arrows. The subset of *discrete inhibitory*

arcs is A_h: $(P_D \times T) \times IN_{+}^{|P|} \to IN_{+}$ or *continuous inhibitory* arcs A_h: $(P_C \times T) \times IR^{|P|} \to IR$. These arcs are drawn with a small circle at the end. The subset A_c defines the *continuous flow* arcs A_c: $((P_C \times T_C) \cup (T_C \times P_C)) \times IR^{|P|} \to IR$, and these arcs are drawn as double arrows to suggest a pipe. A *test* input arc A_t is directed from a place of any kind to a transition of any kind, that A_t: $(P_D \times T) \times IN_{+}^{|P|} \to IN_{+}$ or $(P_C \times T) \times IR^{|P|} \to IR$ and are drawn as dotted single arrows. It does not consume the content of the source place. The arc of net is drawn if the cardinality is not identically zero and it is labeled next to the arc with a default value being 1. The IZ, IN_{+} and IR are the set of discrete integer, natural and real numbers, respectively;

- $K_p : P_D \to IN_{+}$ is the function-capacity of discrete places and for each $p_i \in P_D$ this is represented by the minimum $K_{p_i}^{\min}$ and maximum capacity $K_{p_i}^{\max}$, $0 \le K_{p_i}^{\min} \le K_{p_i}^{\min} < +\infty$ which can contain an *integer* number of *tokens*, respectively. By default $K_{p_i}^{\min} = 0$ and $K_{p_i}^{\max}$ being infinite value;
- $K_b : P_C \to IR$ is the function-capacity of continuous places and for each $p_i \in P_C$ describes the fluid lower bound x_i^{min} and upper bounds x_i^{max} of fluid on each continuous place, that $-\infty < x_i^{\min} < x_i^{\max} < +\infty$. This x_i^{max} by default it is ∞, and bound has no effect when it is set to infinity. Each continuous place has an implicit lower bound at level is 0;
- $G : T \times Bag(P) \to \{True,\ False\}$ is the *guard function* defined for each transition. For $t \in T$ a guard function $g(t, M)$ will be evaluated in each marking M, and if it evaluates to *true*, the transition may be enabled, otherwise t is disabled (by default is *true*);
- *Pri*: $T_D \to IN_{+}$ defines the priority functions for the firing of each transition. By default it is 1. The enabling of a transition with higher priority disables all the lower priority transitions;
- $l : T \cup P \to L$, is a labeling function that assigns a label to a nodes (transitions and places) of net. In this way that maps the node name of net into action name or in condition name that $l(t_j) = l(t_k) = \alpha$ but $t_j \neq t_k$ or $l(p_i) = l(p_n) = \iota$ but $p_i \neq p_n$, respectively. ■

The structure of a *GDPN* is static. Assuming that the behaviour of the system can be described in terms of the current system state and its possible changes, the dynamics of a net structure is specified by defining its marking and marking evolution rule.

Definition 2: A timed marked labeled *GDPN* is a pair $NH = \langle N, M_0 \rangle$, where $N = \langle H\Gamma, \theta, W, V \rangle$ is a labeled *GDPN* structure (see Definition 1) with the respectively attributes of timed transitions and M_0 is the initial marking of the net such as:

- The current marking (state) value of a net depends on the kind of place, and it is described by a pair of vector-columns $M = (\boldsymbol{m}, \boldsymbol{x})$, where the $\boldsymbol{m}$: $P_D \to IN_{+}$ and $\boldsymbol{x}$: $P_C \to IR$ are marking functions of respectively type of places. The vector-column $\boldsymbol{m} = (m_i p_i,\ m_i \ge 0, \forall\, p_i \in P_D)$ with $m_i p_i$ is the number $m_i = \boldsymbol{m}(p_i)$ of tokens in discrete place, and it is represented by the black dots. The $x = (x_k b_k,\ x_k \ge x_k^{\min}, \forall\, b_k \in P_C)$ is vector-column, where $x_k b_k$ is the fluid level $x_k = \boldsymbol{x}(b_k)$ in continuous place b_k, and it is the real number, that is allowed to take also *negative* real value. The initial marking of net is $M_0 = (\boldsymbol{m}_0, \boldsymbol{x}_0)$. The vector $\boldsymbol{m}_0$ gives the initial marking of discrete places and the vector $\boldsymbol{x}_0$ gives the initial marking of fluid places;
- The set of discrete transitions T_D is partitioned into $T_D = T_0 \cup T_\tau, T_0 \cap T_\tau = \varnothing$ so that: T_τ is a set of timed discrete transitions and T_0 is a set of immediate discrete transitions. The $Pri(T_0) > Pri(T_\tau)$. A timed discrete transition $t \in T_\tau$ is drawn as a black rectangle and has a firing delay $\theta : T_\tau \times Bag(P) \to IR_{+}$ is associated to it, and this is can be marking dependent. The IR_{+} is the set of nonnegative real numbers.

Let $T(M)$ denote the set of enabled transitions in current marking $M = (\boldsymbol{m}, \boldsymbol{x})$. Thus, a timed transitions $t \in T_\tau(M)$ is enabled in current tangible marking M, it fires after delay $\theta(t, M)$. By default it is 1. Note once again, we do allow the firing delay to be dependent on fluid levels;

- $W : T_0 \times Bag(P) \to IR_{+}$ is the weight function of immediate discrete transitions $t_j \in T_0$, and this type of transition is drawn with a black thin bar and has a zero constant firing time. If several enabled immediate transitions $t_j \in T_0(T)$ are scheduled to fire at the same time in *vanishing* marking M, the transitions t_k with the respective weights w_k fire with probability:

$$q(t_k, M) = w(t_k, M) / \sum_{t_j \in T_0(M)} w(t_j, M);$$

- $V : T_c \times Bag(P) \to IR$ is the marking dependent fluid rate function of timed continuous transitions T_c. These rates appear as labels next to the continuous timed transitions. By default it is 1. If $t_i \in T_c$ is enabled in *tangible* marking *M* it fires with rate $V_i(M)$, that continuously change the fluid level of continuous place P_C. ■

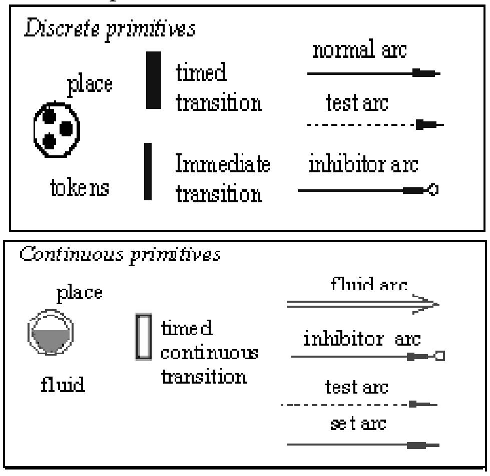

Figure 1. All the primitive of the timed *GDPN*.

Figure 1 summarizes the graphical representation of all the *GDPN* primitives.

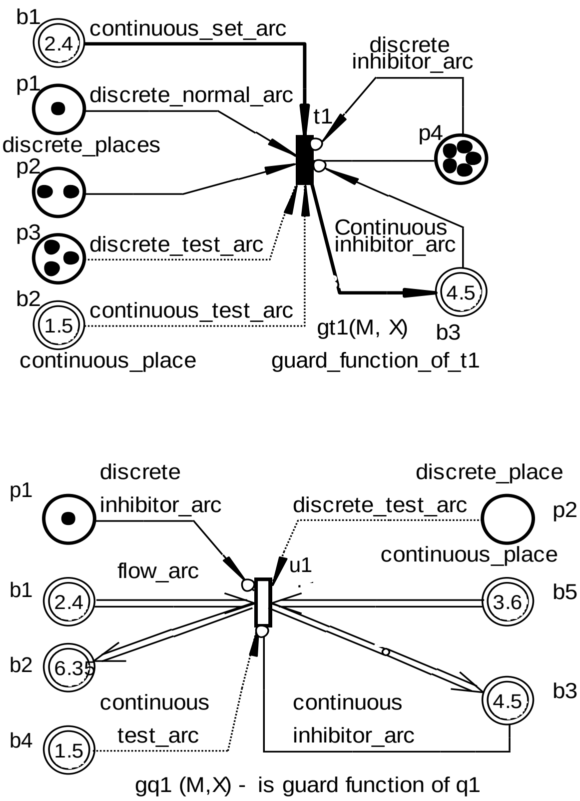

Figure 2. All the possible ways of placing arcs in a timed *GDPN* net.

Figure 2 summarize the all possible ways of placing arcs in a *GDPN* net for discrete transition and continuous transition with the discrete places and continuous places, respectively.

Upon firing, the discrete (continuous) transition removes a specified number (quantity) of tokens (fluid) for each discrete (continuous) input place, and deposits a specified number (quantity) of tokens (fluid) for each discrete (continuous) output place. The fluid levels of continuous places can change the enabling/disabling of discrete and continuous transitions.

The role of the previous set and functions will be clarified by providing the enabling and firing rules. Let us denote by m_i the *i*-th component of the vector ***m***, i.e., the number of tokens in discrete place p_i when the marking is ***m***, (and x_k denote the *k*-th component of the vector ***x***, i.e., the fluid level in continuous place p_k).

Enabling and Firing of Transitions. Let *T*(*M*) the set o enabled transitions in current marking *M*. We say that a discrete transition $t_j \in T_D(M)$ is enabled in current marking *M* if the following enabling condition $ec_d(t_j)$ is verified:

$$ec_d(t_j) = \Big(\bigwedge_{\forall p_i \in {}^\bullet t_j} (m_i \geq \mathrm{Pre}(p_i, t_j)\Big) \;\&\; \Big(\bigwedge_{\forall p_k \in {}^\circ t_j} (m_k < Inh(p_k, t_j)\Big) \;\&\; \Big(\bigwedge_{\forall p_l \in {}^* t_j} (m_l \geq Test(p_l, t_j)\Big) \;\&\; \Big(\bigwedge_{\forall p_n \in t_j^\bullet} ((K_p - m_n) \geq Post(p_n, t_j)\Big) \;\&\; \Big(\bigwedge_{\forall b_i \in {}^\bullet t_j} (x_i \geq \mathrm{Pr}e(b_i, t_j)\Big) \;\&\; \Big(\bigwedge_{\forall b_k \in {}^\circ t_j} (x_k < Inh(b_k, t_j)\Big) \;\&\; \Big(\bigwedge_{\forall b_l \in {}^* t_j} (x_l \geq Test(b_l, t_j)\Big) \;\&\; g(t_j, M) \;\&\; \Big(\bigwedge_{\forall b_n \in t_j^\bullet} ((K_b - x_n) \geq Post(x_n, t_j)\Big). \qquad (2.1)$$

The discrete transition $t_j \in T_D(M)$ fire if no other discrete transition $t_k \in T_D(M)$ with higher priority has enabled.

Also, we say that a continuous transition $u_j \in T_C(M)$ is enabled and continuously fire in current marking *M* if the enabling condition $ec_c(u_j)$ is verified:

$$ec_c(u_j) = \Big(\bigwedge_{\forall b_i \in {}^\bullet u_j} (x_i > 0)\Big) \;\&\; \Big(\bigwedge_{\forall p_k \in {}^\circ u_j} (m_k < Inh(p_k, u_j)\Big) \;\&\; \Big(\bigwedge_{\forall p_l \in {}^* u_j} (m_l \geq Test(p_l, u_j)\Big) \;\&\; \Big(\bigwedge_{\forall b_k \in {}^\circ u_j} (x_k < Inh(b_k, u_j)\Big) \;\&\; g(t_j, M) \;\&\; \Big(\bigwedge_{\forall b_l \in {}^* u_j} (x_l \geq Test(b_l, u_j)\Big) \;\&\; \Big(\bigwedge_{\forall b_n \in u_j^\bullet} ((K_b - x_n) \geq V_j \cdot Post(x_n, u_j)\Big), \qquad (2.2)$$

and no other continuous transition with higher priority has concession.

The current marking M evolves in time τ. If an immediate discrete transition has concession in current marking $M=(\boldsymbol{m}, \boldsymbol{x})$, it is enabled and the marking is *vanishing*. Otherwise, the marking is *tangible* and any timed discrete transition with concession is enabled in it. An immediate discrete transition t_j enabled in current marking $M=(\boldsymbol{m}, \boldsymbol{x})$ yields a new vanishing marking $M' = (\boldsymbol{m}', \boldsymbol{x})$. We can write $(\boldsymbol{m}, \boldsymbol{x})$ $[t_j > (\boldsymbol{m}', \boldsymbol{x})$). Thus, the new discrete marking is:

$$\boldsymbol{m}'(p_i) = \boldsymbol{m}'(p_i) + Post(\cdot, p_i) - \Pr e(\cdot, p_i).$$

If the $M=(\boldsymbol{m}, \boldsymbol{x})$ is tangible, the fluid could continuously flow through the flow arcs A_c of enabled continuous transitions into or out of fluid places. As a consequence of this, a continuous transition t_c is *enabled* at M iff for every $p_c \in {}^{\bullet}t_c$, $\boldsymbol{x}(p_c)>0$, and its enabling degree *enab*(t_c, M) is [1]:

$$enab(t_c, M) = \min_{p_c \in {}^{\bullet}t} \{\boldsymbol{x}(p_c)/Pre(p_c, t_c)\}.$$

In the *GDPN* the current dynamic balance $\beta_k(M)$ change the fluid level of continuous place $p_k = P_C$ in current *tangible* marking M is given by the following relation:

$$\beta_k(M) = \beta_k^+(M) - \beta_k^-(M), \text{ with}$$

$$\beta_k^+(M) = \sum_{\forall t_j \in T_C(M)} \mathrm{V}_j(M) \cdot Pre(t_j, b_k) \text{ and}$$

$$\beta_k^-(M) = \sum_{\forall t_l \in T_C(M)} \mathrm{V}_k(M) \cdot Post(t_l, b_k),$$

where for any $t_l \in T_C$ the $\beta_k^+(M)$ is an input fluid rate of continuous place $p_k = P_C$, $\beta_k^-(M)$ is an output fluid rate of this place, and $T_c(M)$ is a set of continuous transition teach which is enabled in M.

Thus, during the time $\Delta\tau \to 0$ the fluid level $x_k = \boldsymbol{x}(b_k)$ continuously change an conformity with the relation: $x_k' = x_k + \beta_k(M) \cdot \Delta\tau$.

We allow the firing rates and the enabling functions of the timed discrete transitions, the firing speeds and enabling functions of the timed continuous transitions, and arc cardinalities to be dependent on the current state of the *GDPN*, as defined by the marking M.

Let the dynamic balances $\beta_i(M)$ which change levels for each continuous place $p_i \in P_C$ in current tangible marking M_k are collected into diagonal matrices:

$$\boldsymbol{\beta}(M_k) = diag(\beta_1(M_k), \ldots, \beta_i(M_k), \ldots, \beta_n(M_k)),$$
$$i = 1, \ldots, n, \; n = |P_c|.$$

The pair $s_k = (M_k, \; \beta(M_k))$ describes the current state of the *GDPN*. Let we denote state space of the net by $S = IN_+^{|P_D|} \times IR^{|R_C|}$, that $S = S_D \cup S_C$, $S_D \cap S_C = \varnothing$, where S_D denote the discrete components and S_C denote the continuous components of the S. The marking process of net at time τ is:

$$S(\tau) = \{ \; s(\tau), \; \tau \geq 0 \; \}.$$

2. DYNAMIC REWRITING GDPN

In this section we introduce the model of *descriptive dynamic rewriting NH* system.

The approach proposed in [5, 6] consists in incorporating compositional features into *GDPN* models. In this way the modeler can identify what will constitute a basic component and can build the model with the use of *descriptive operations*.

Due to the space restrictions we will only give a brief overview to this topic and refer the reader to [5] and the references therein.

Let $X \rho Y$ is a binary relation. The *domain* of ρ is the *Dom*$(\rho) = \rho Y$ and the *codomain* of ρ is the *Cod*$(\rho) = X\rho$ [7].

Let $A = < Pre, Post, Test, Inh >$ is a set of arcs belong to net $H\Gamma = < P, T, Pre, Post, Test, Inh, K_p, K_b, G, Pri, l >$ (see *Definition* 1).

Definition 3**.** A *dynamic rewriting NH* net is a system $RN = < N, \; R, \phi, G_{tr} \; G_r, \; M >$, where:

- $N = < H\Gamma, \; \theta, W, V >$ and $R = \{r_1, \ldots, r_k\}$ is a finite set of discrete rewriting rules (DR) about the runtime structural modification of net that $P \cap T \cap R = \varnothing$. In the graphical representation, the DR rule is drawn as a two embedded empty rectangle. We denote the set of *events* of the net by $E = T_D \cup R$;
- $\phi : E \to \{T_D, R\}$ is a function indicate for every rewriting rule the type of event can occur;
- $G_{tr} : R \times Bag(P) \to \{True, False\}$ is the *transition rule guard function* associated with $r \in R$, and $G_r : R \times Bag(P) \to \{True, \; False\}$ is the *rewriting rule guard function* defined for each rule of $r \in R$, respectively. For $\forall r \in R$, the function $g_{tr}(M) \in G_{tr}$ and $g_r(M) \in G_r$ will be evaluated in each marking and if its are evaluates to *True*, the rewriting rule r may be *enabled*, otherwise it is *disabled*. Default value of $g_{tr}(M) \in G_{tr}$ is *True* and for $g_r(M) \in G_r$ is *False*) in current marking M.

Let the $R\Gamma = < N,\ R, \phi, G_{tr}\ G_r >$ and the $RN = < R\Gamma,\ M >$ are represented by the descriptive expression $DE_{R\Gamma}$ and DE_{RN}, respectively. A dynamic rewriting structure modifying rule $r \in R$ of *RN* is a map $r : DE_L \rhd DE_W$, where whose *codomain* of the $\rhd$ *rewriting operator* is a fixed descriptive expression DE_L of a subnet RN_L of current net *RN*, where $RN_L \subseteq RN$ with $P_L \subseteq P$, $E_L \subseteq E$ and the set of arcs $A_L \subseteq A$, and whose *domain* of the $\rhd$ is a descriptive expression DE_W of a new RN_w subnet with $P_w \subseteq P$, $E_w \subseteq E$ and set of arcs A_W.

The rewriting operator $\rhd$ represent binary operation which produce a *structure change* in the DE_{RN} and the net *RN* by replacing (rewriting) of the fixed current DE_L of subnet RN_L (DE_L and RN_L are dissolved) by the new DE_W of subnet RN_W now belong to the new modified resulting $DE_{RN'}$ of net $RN' = (RN \setminus RN_L) \cup RN_W$ with $P' = (P \setminus P_L) \cup P_W$ and $E' = (E \setminus E_L) \cup E_W$, $A' = (A - A_L) + A_W$ where the meaning of \ (and $\cup$) is operation to removing (adding) RN_L from (RN_W to) net *RN*. In this new net RN', obtained by execution (fires) of enabled rewriting rule $r \in R$, the places and events with the same attributes which belong RN' are fused. By default the rewriting rules $r : DE_L \rhd \varnothing$ or $r : \varnothing \rhd DE_W$ describe the rewriting rule which fooling holds the $RN' = (RN \setminus RN_L)$ or the $RN' = (RN \cup RN_W)$.

A state configuration of a net *RN* is a pair ($R\Gamma, s$), where $R\Gamma$ is the current structure of net together with a current state s, $s = (M,\ \beta(M))$. The ($R\Gamma_0, s_0$) with $P_0 \subseteq P, E_0 \subseteq E$ and state s_0 is called the initial state configuration of a net *RN*. ■

Enabling and Firing of Events. The enabling of events depends on the marking of all places. We say that a transition $t_j \in T_D$ of event e_j is enabled in current marking *M* if the if the enabling condition $ec_d(t_j, M)$ described by the logic expression (2.1).

The discrete rewriting rule $r_j \in R$, that change the structure of *RN*, is enabled in current marking *M* if the following enabling condition $ec_{tr}(r_j, M)$ is verified:

$$ec_{tr}(r_j, M) = (\underset{\forall p_i \in {}^\bullet r_j}{\wedge} (m_i \geq \Pr e(p_i, r_j))\ \&$$

$$\underset{\forall p_k \in {}^\circ r_j}{\wedge} (m_k < Inh(p_k, r_j))\ \&$$

$$\underset{\forall p_l \in {}^* r_j}{\wedge} (m_l \geq Test(p_l, r_j))\ \&$$

$$\underset{\forall p_n \in r_j^\bullet}{\wedge} ((K_{p_n} - m_n) \geq Post(p_n, r_j))\ \&$$

$$(\underset{\forall b_i \in {}^\bullet r_j}{\wedge} (x_i \geq \Pr e(b_i, r_j))\ \&\ (\underset{\forall b_k \in {}^\circ r_j}{\wedge} (x_k < Inh(b_k, r_j))\ \&\ (\underset{\forall b_l \in {}^* r_j}{\wedge} (x_l \geq Test(b_l, r_j))\ \&$$

$$g(r_j, M)\ \&\ g_{tr}(r_j, M))\ .$$

Let the $T_D(M)$ and the $R(M)$ that $T_D(M) \cap R(M) = \varnothing$ is the set of enabled discrete transitions and rewriting rule in current marking *M*, respectively. We denote the set of enabled events in a current marking *M* by $E(M) = T_D(M) \cup R(M)$.

The event $e_j \in E(M)$ fire if no other event $e_k \in E(M)$ with higher priority has enabled. Hence, for each event e_j *if* $((\phi_j = t_j) \vee (\phi_j = r_j) \wedge (g_{tr}(r_j, M) = False))$ *then* (the firing of transition $t_j \in T_D(M)$ or rewriting rule $r_j \in R(M)$ change only the current marking: $(R\Gamma, s) \xrightarrow{e_j} (R\Gamma, s') \Leftrightarrow (R\Gamma = R\Gamma$ and the $M[e_j > M'$ in $R\Gamma$)). Also, for e_j event *if* $((\phi_j = r_j) \wedge (g_r(r_j, M) = True))$ *then* (the event e_j occur to firing of rewriting rule r_j and it change the configuration and marking of current net in following way:

$$(R\Gamma, s) \xrightarrow{r_j} (R\Gamma', s'),\ M[r_j > M').$$

The accessible state graph of a $RN = < R\Gamma, M >$ net is the labeled directed graph whose nodes are the states and whose arcs which is labeled with events or rewriting rules of *RN* that are of two kinds:

a) *firing* of a enabled $e_j \in E(M)$ event determine the arc from the state $(R\Gamma, s)$ to the state ($R\Gamma, s'$) which is labeled with event e_j then this event can fire in the net configuration $R\Gamma$ at marking *M* and leads to new state such as s': $(R\Gamma, s) \xrightarrow{e_j} (R\Gamma', s') \Leftrightarrow (R\Gamma = R\Gamma'$ and $[M[e_j > M'$ in $R\Gamma$);

b) *change configuration*: arcs from state $(R\Gamma, s)$ to state ($R\Gamma', s'$) labeled with the rewriting rule $r_j \in R$, that $r_j : (R\Gamma_L, M_L) \rhd (R\Gamma_W, M_W)$ which represent the change configuration of current *RN* net: $(R\Gamma, s) \xrightarrow{r_j} (R\Gamma', s')$ with $M[r_j > M'$.

As en example, let we consider the discrete part *RN*1 net given by the following descriptive expression:

$$DE_{R\Gamma 1} = p_1 |_{r_1} p_2 \vee \tilde{p}_1 |_{u_1} b_1[2.75] \vee DE'_{R\Gamma 1},$$

$$DE'_{R\Gamma 1} = (p_2 \cdot p_5 \cdot b_1)|_{t_1} p_3 |_{t_2} p_4 |_{t3} (p_1 \Diamond p_5) \vee DE''_{R\Gamma 1},$$

$$DE''_{R\Gamma 1} = (p_5 \cdot b_1[1.8])|_{u_2} b_2 \vee (p_4 \cdot b_1)|_{u_3},$$

$$M_0 = (5p_1, 1p_5, 12.5x_1),$$

$$\beta(M_0) = diag(0.95, 0), \quad r_1 : DE_{R\Gamma 1} \triangleright DE_{R\Gamma 2},$$

$$g_r(r_1, M) = (m_1 = 3) \& (m_5 = 0).$$

Also, for rewriting rule r_j is required to identify if RN_L net belong the $R\Gamma$. Upon firing, the enabled events or rewriting rule modify the current marking and/or and modify the structure and current marking of *RN*1 in *RN*2 given by the following descriptive expression:

$$DE_{R\Gamma 2} = p_1 |_{t_1} p_2 \vee \tilde{p}_1 |_{u_1} b_1[2.75] \vee DE'_{R\Gamma 2},$$

$$DE'_{R\Gamma 2} = (p_2 \cdot p_6 \cdot b_1[1.5])|_{t_2} p_3(|_{t_3} p_4 |_{t_4} p_5 \vee$$

$$|_{t_5} p_5 |_{r_2} (p_1 \Diamond p_6)) \vee DE''_{R\Gamma 2},$$

$$DE''_{R\Gamma 1} = b_1 |_{u_2} b_2 \vee (b_2 \cdot b_3)|_{u_3} \vee \tilde{p}_4 |_{u4} b_3,$$

$$M = (1p_1, 3p_2, 1p_3, 15.8b_3),$$

$$\beta(M) = diag(1.75, 0, 0), \quad r_2 = r_1^{-1} : DE_{R\Gamma 2} \triangleright DE_{R\Gamma 1},$$

$$g_r(r_2, M) = (m_1 = 4) \& (m_5 = 1).$$

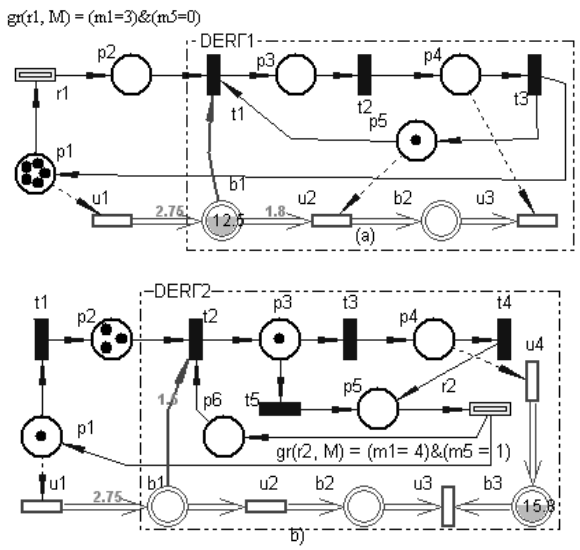

Figure 3. Translation of (a) $DE_{R\Gamma 1}$ in *RN*1 and (b) $DE_{R\Gamma 2}$ in *RN* 2.

The translation of $DE_{R\Gamma 1}$ in *RN*1 is shown in figure 4a, and $DE_{R\Gamma 2}$ in *RN* 2 is shown in figure 4b.

3. CONCLUSIONS

In this paper we have defined the dynamic rewriting *GDPN* models from the behavioral state based process run-time structure change of system components. This approach can preserve the functional structure of the model and support several behavioral types between constituent components.

However, this approach is still interesting as a tool to combining the visual simulation at run-time structure change of *GDPN* for performance discrete-continuous modeling of computer systems.

Bibliography

1. ***Alla, A., David, H.*** Continuous and hybrid Petri nets. Journal of Systems Circuits and Computers, 8 (1) , 1998, pp. 159-188.
2. ***Ajmone-Marsan, M., Balbo, G., Chiola, G., Donatelli, S., Francheschinis, G.*** Modeling with Generalized Stochastic Petri Nets// John Wiley&Sons, 1995.
3. ***David, R.*** Modelling Hybrid Systems using Continuous and Hybrid Petri Nets. In Proceedings of International Conference PNPM'97, 1997, pp.135-144.
4. ***Demongodin, I., Koussoulas, N.T.*** Differential Petri Nets: Representing continuous systems in a discrete-event world. IEEE Transactions on Automatic Control, Vol. 43, No. 4, 1998.
5. ***Guţuleac, E.*** Descriptive Compositional Construction of GSPN Models for Performance Evaluation of Computer Systems. In: Proc. of 8-th International Symposium on Automatic Control and Computer Science, SAACS2004, Iasi, România, 22-23 October 2004.
6. ***Guţuleac, E.*** Descriptive self-rewriting generalized stochastic Petri nets for modeling of computer systems. Meridian Ingineresc, Nr. 2, 2005, pp.18-23.
7. ***Llorens, M. and Oliver, J.*** Structural and Dynamic Changes in Concurrent Systems: Reconfigurable Nets. In: IEEE Transactions on Computers, vol. 53, no. 9, pp. 1147-1158, September 2004.
8. ***Murata, T.*** Petri Nets: Properties, Analysis and Applications. In Proceeding of the IEEE, vol.77, no. 4, 1989, pp.541-580.

Recomandat spre publicare*: *21.02.06