ENCODING P SYSTEMS WITH DESCRIPTIVE MEMBRANE TIMED PETRI NETS

Emilian Guţuleac

Technical University of Moldova, Computer Science Department
E-mail: egutuleac@mail.utm.md

INTRODUCTION

Petri nets (PN) are very popular formalism for the analysis and representation of concurrent distributed systems that has draw much attention from the community to modeling and verification of this type of systems [1].

P systems, also referred to as membrane systems, are a class of parallel and distributed computing models inspired from the structure and the functioning of living cells [2]. The interest of relating P systems with the PN model of computation lead to several important results on simulation and decidability issues.

Some efforts have been made to simulate P systems with Petri nets [3].

In this paper, we introduce the *Descriptive Membrane Timed* PN, called DM-nets that can dynamically modify their own structures by rewriting rules transitions some of their components thus supporting structural dynamic changes within modeled P systems.

DESCRIPTIVE DYNAMIC REWRITING TIMED PETRI NETS

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 $Cod(\rho)=X\rho$. Let $A=<Pre, Post, Test, Inh>$ is a set of arcs belong to a labeled composite Petri net $\Gamma=<P, T, Pre, Post, Test, Inh, G, Pri, K_p, l>$ definite in [1].

Definition 1. A *descriptive dynamic rewriting* PN system is a structure $RN=<\Gamma, R, \phi, G_r, M>$, where: $R=\{r_1, ..., r_k\}$ is a finite set of rewriting rules about the runtime structural modification of net. Let $E=T\cup R$ denote the set of *events* of the net; $\phi: E\rightarrow\{T, R\}$ is a function indicate for every transition the type of event can occur; $G_r: T\times IN_+^{|P|}\rightarrow\{TRUE, FALSE\}$ is the *rewriting rule guard function* defined for each rule *r* of transition *t*. For $r\in R$, $g_r(r, M)\in G_r$ is a Boolean function that will be evaluated in each marking, and if it evaluates to *TRUE*, the rewriting rule *r* of transition *t* may be *enabled*, otherwise it is *disabled* (default value is *FALSE*). Let $R\Gamma = < \Gamma, R, \phi, G_r >$ and $RN = < R\Gamma, M >$ with 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 $\rhd$ 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 set of arcs $A_L \subseteq A$ and whose *domain* of $\rhd$ is a descriptive expression DE_W subnet of a new RN_w with $P_w \subseteq P$ and $E_w \subseteq E$ and set of arcs A_W. The $\rhd$ operator 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 is 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, respectively. By default the rewriting rules $r: DE_L \rhd \varnothing$ and $r: \varnothing \rhd DE_W$ respectively describe the rewriting rule which fooling holds $RN' = (RN \setminus RN_L)$ and $RN' = (RN \cup RN_W)$.

A state of a net *RN* is a pair ($R\Gamma, M$), where $R\Gamma$ is the configuration of net together with a current marking *M*. Also, the pair ($R\Gamma_0, M_0$) with $P_0 \subseteq P$, $E_0 \subseteq E$ and marking M_0 is called the initial state of the net. ■

Enabling and Firing of Events. The enabling of events depends on the marking of all places. The transition t_j of event e_j is enabled in current marking *M* if the enabling condition $ec(t_j, M)$ is verified [ ]. The rewriting rule $r_j \in R$ is enabled in current marking *M* if is verified the following enabling condition: $ec(r_j, M) = ec(t_j, M)$ & $g_r(r_j, M)$.

Let *E*(*M*) is the set of enabled events in a current marking *M*. The event $e_j \in E(M)$ fire if no other event $e_k \in E(M)$ with higher priority has enabled. Also, *if* ($g_r(r_j, M)$) *then* ($\phi_j = r_j$, the event occur to rewriting rule r_j and its occurrence change configuration and marking of current net: $(R\Gamma, M) \xrightarrow{t_j} (R\Gamma', M')$) *else* ($\phi_j = t_j$, the event occur to t_j and it change the current marking: $(R\Gamma, M) \xrightarrow{r_j} (R\Gamma', M') \Leftrightarrow (R\Gamma = R\Gamma'$ and $M[t_j > M'$ in $R\Gamma$).

The state graph of a net $RN = < \Gamma, M >$ is the labeled directed graph whose nodes are the states and whose arcs which is labeled with events of RN are of two kinds: a) firing of a enabled transition

$t \in E(M)$: arcs from state (Γ, M) to state(Γ', M') labeled with transition t then this transition can fire in the net configuration Γ at marking M and leads to new marking $M' : (\Gamma, M) \xrightarrow{t} (\Gamma', M') \Leftrightarrow (\Gamma = \Gamma'$ and $M\,[t > M'$ in Γ); b) change configuration: arcs from state (Γ, M) to state(Γ', M') labeled with rewriting rule $r \in R$, $r : (\Gamma_L, M_L) \triangleright (\Gamma_W, M_W)$ which represent the change configuration of current *RN*.

Systems are described in timed PN (TPN) as interactions of components that can performed a set of activities associated with events. An event $e = (\alpha, \theta)$, where $\alpha \in E$ is the type of the activity (action name), and θ is the firing delay.

Definition 2. A descriptive dynamic rewriting TPN as a $RTN = < RN,\ \theta >$, where:

- $RN = < \Gamma,\ R, \phi, G_r,\ M >$; $\Gamma = < P, T, Pre, Post, Test, Inh, G, Pri, M_0 >$ (see Def. 2 and 3) with set of events E which can be partitioned into a set E_0 of *immediate* events and a set E_τ of *timed* events, $E = E_0 \cup E_\tau$, $E_0 \cap E_\tau = \varnothing$. The immediate event is drawn as a black thin bar and timed event is drawn as a black rectangle, and $Pri(E_0) > Pri(E_\tau)$;
- $\theta : E \times IN_+^{|P|} \to IR_+$ is the weight function that maps events onto real numbers IR_+ (delays or weight speeds). Its can be marking dependent. The delays $\theta(e_k, M) = d_k(M)$ defining the events firing parameters governing its duration for each timed events of E_τ. If several timed events are enabled concurrently $e_j \in E(M)$ for $e_j \in {}^\bullet p_i = \{\forall e_j \in E : \Pr e(p_i, e_j) > 0\}$, either in competition or independently, we assume that a *race condition* exists between them. The evolution of the model will determine whether the other timed events have been aborted or simply interrupted by the resulting state change.

The $\theta(e_j, M) = w_j(M)$ is weight speeds of immediate events $e_j \in E_0$. If several enabled immediate events are scheduled to fire at the same time in *vanishing* marking M with the weight speeds, and the probability to enabled immediate event e_j can *fire* is:

$$q_j(M) = w(e_j, M) / \sum_{e_l \in (E(M) \& {}^\bullet p_i)} w(e_l, M), \text{ where}$$

$E(M)$ is the set of enabled events in M. An immediate events $e_j \in T_0$ has a zero firing time. ■

DESCRIPTIVE MEMBRANE TIMED PETRI NETS

Here we give a brief review of P systems. A full guide for P systems can be referred to [3]. In general, a basic evolution-communication P system with active membranes (of degree $m \geq 0$) is $\Pi = (O, L, \mu, \Omega, (\rho, \pi))$, where: O alphabets of objects; L is a finite set of labels for membranes; μ is a membrane structure consisting of m membranes $[_i\ \ ]_i$ labeled with elements i in L; Ω is the configuration , that is a mapping from membranes of Π (nodes in μ) to multisets of objects $\omega_j \in \Omega$, j=1,…, $|\Omega|$ from O; ρ and π is respectively the set off developmental rules ρ_i and its priorities π_i , i= 0,1,…, m-1. Thus can be of two forms: *object rules* (OR), i.e., evolving and communication rules concerning objects, and rewriting *membranes rules* (MR), i.e., the rules about the structural modification of membranes.

In this section we define *DME-Nets* an encoding of P systems mentioned above into descriptive dynamic rewriting TPN as a *RTN*. The basis for *DME-Nets* is a membrane *RTN's* that is *DE* net structure comprise: places; transitions; weighed directed arcs from places to transitions and vice-versa; a capacity for each place; weighed inhibitory and test arcs; priority and guard function of transitions.

Consider the P system Π . The encoding of Π into RTN_{Π} is decomposed into two separate steps. First, for every object $\omega_j \in \Omega$ in membrane $[_i \ ]_i$ is one-to-one mapped to a place $p_{i,j}=[_i \ m_j^0 p_j \ ]_i$ labeled as ω_j with the initial marking m_j^0, and each rule $\rho_{i,k} \in \rho$ is one-to-one mapped to a event $e_{i,k} = [_i \ e_k \ ]_i$ labeled as $\rho_{i,k}$ that acts on the this membrane. Also, to represent the configuration of a membrane $[_i \ ]_i$ in Π as we associate a descriptive expression DE_i that represent a RTN_i which corresponds to transformation, communication and rewriting *membranes rules*, where is rules that are purely local to one membrane that it correspond to the initial configuration of the P system Π as $[_i DE_i]_i$. By construction, the places and events used in the DE_i are all distinct.

Second, the nets RTN_i are merged with special transition (events) in order to code communication rules in Π . Thereby, the net RTN_{Π} of Π is obtained by merging the nets RTN_i, that is represented by the $DE_{\Pi} = \vee_{i=1}^{n} DE_i$, extended with one transition for every communication rules in Π .

Let u, v and u', v' is a multiset of objects. The *evolving* object rule $\rho_{h',j} : [_h \ [_{h'} \ u \to v \ ]_{h'} \ ]_h$ with multiset of objects u, v which will be kept in membrane h' is encoded as $[_h \ [_{h'} \ m_u p_u \ |_{t_j} \ m_v p_v \ ]_{h'} \ ]_h$ that m_u and m_v represent the current marking of places $p_{h',u}$ and $p_{h',v}$, respectively. The antiport rule $\rho_{h',i} : [_h u \ [_{h'} \ v \ ]_{h'} \ ]_h \to [_h v' \ [_{h'} \ u' \ ]_{h'} \ ]_h$, that realize a synchronized with object c the exchange of objects, is encoded as $[_h \ [_{h'} \ (p_u \cdot p_v \cdot \tilde{p}_c) |_{t_i} \ (p_{u'} \Diamond p_{v'}) \ ]_{h'} \ ]_h$. Also, the $\rho_{h',k} : [_h u \ [_{h'} \ ]_{h'} \ ]_h \to [_h \ [_{h'} \ u' \ ]_{h'} \ ]_h$ symport rule that move objects from inside to outside a membrane, or vice-versa is encoded as $[_h \ [_{h'} \ (p_u \cdot \tilde{p}_c) |_{t_k} \ p_{u'}) \ ]_{h'} \ ]_h$. Because a configuration mean both a membrane structure and the associated multisets, we need rules for processing membranes and multisets of objects as: *MR* ={*Change, Dissolve, Create, Divide, Merge, Separate, Move*}.

The above membrane rewriting rules (realized by the rewriting events in *DE's*) are defined as follows: *Change* $[_h\,[_{h'}\,(DE_{h'}, M_{h'})\,]_{h'}\,]_h \rhd [_h\,[_{h'}\,(DE'_{h'}, M'_{h'})\,]_{h'}\,]_h$ that in runtime the current structure and the multisets of objects to membrane *h*, encoded by descriptive expression $DE_{h'}$ and marking $M_{h'}$ is changed in a new structure $DE'_{h'}$ with new marking $M'_{h'}$; *Dissolve* rewriting rule $[_h\,[_{h'}\,(DE_{h'}, M_{h'})\,]_{h'}\,]_h \rhd [_h\,\,(DE_{h'}, M'_{h'})\,\,]_h$ that the objects and sub-membranes of membrane h' now belong to its parent membrane h, the skin membrane cannot be *dissolved*; *Create* rewriting rule $[_h\,\,(DE_h, M_h)\,\,]_h \rhd [_h\,(DE'_h, M'_h)\,\,[_{h'}\,(DE''_{h'}, M''_{h'})\,]_{h'}\,]_h$ with $M_h = M'_h + M''_{h'}$ that the new membrane h' is created and $M''_{h'}$ are added into membrane h', the rest remain in the parent membrane *h*; *Divide* rewriting rule $[_h\,\,(DE_h, M_h)\,\,]_h \rhd [_h\,[_{h'}\,(DE_h, M_h)\,]_{h'}\,[_{h''}(DE_h, M_h)\,\,]_{h''}]_h$ that the objects and sub- membranes are reproduced and added into membrane h' and membrane h'', respectively; *Merge* rewriting rule that the objects of membrane h' and h'' are added to a new membrane h is $[_h\,[_{h'}\,(DE'_{h'}, M_{h'})\,]_{h'}\,[_{h''}(DE''_{h''}, M''_{h''})\,\,]_{h''}]_h \rhd [_h\,\,(DE'_{h'} \vee DE''_{h''}, M_{h'} + M''_{h''})\,\,]_h$. *Separate* rewriting rule is the counterpart of *Merge* is done by a rewriting rule of the form $[_h\,(DE'_{h'} \vee DE''_{h''},\; M_{h'} + M''_{h''})\,]_h \rhd [_h\,[_{h'}\,(DE'_{h'}, M_{h'})\,]_{h'}\,[_{h''}(DE''_{h''}, M''_{h''})\,]_{h''}]_h$ with the meaning that the content of membrane h is split into two membranes, with labels h' and h''. *Move* rewriting rule $[_h\,\,[_{h'}\,(DE_{h'}, M_{h'})\,[_{h''}(DE_{h''}, M_{h''})\,]_{h''}]_{h'}]_h \rhd [_h\,[_{h'}\,(DE_{h'}, M_{h'})\,]_{h'}\,[_{h''}(DE_{h''}, M_{h''})\,]_{h''}]_h$ or

$[_h\,[_{h'}(DE_{h'}, M_{h'})\,]_{h'}\,[_{h''}\,(DE_{h''}, M_{h''})\,]_{h''}]_h \rhd [_h\,\,[_{h'}\,(DE_{h'}, M_{h'})\,[_{h''}(DE_{h''}, M_{h''})\,]_{h''}]_{h'}]_h$, where a membrane h'' can be moved out or moved into a membrane h' as a whole.

CONCLUSIONS

We have defined a set of descriptive composition operation and rewriting rules attached with transitions for the creation of dynamic rewriting TPN and descriptive membrane timed Petri nets models from behavioral state based process runtime structure change of P systems.

We are currently implementing a simulator for descriptive rewriting TPN models.

REFERENCES

[1] Guţuleac E. Descriptive Compositional Construction of GSPN Models for Performance Evaluation of Computer Systems. In: Proceeding of 8-th International Symposium on Automatic Control and Computer Science, SACCS2004, Iasi, România, 2004. pp. 43-50.

[2] Gh. Păun. Membrane Computing: An introduction. Springer-Verlag, Berlin, 2002.

[3] Z. Qi, J. You, and H. Mao. P Systems and Petri Nets. In Proc. WMC 2003, Lecture Notes in Computer Science, vol. 2933, Springer, Berlin, 2003, pp. 387-403.