DE GRUYTER OPEN

DOI: 10.1515/aucts-2015-0023

ACTA UNIVERSITATIS CIBINIENSIS – TECHNICAL SERIES
Vol. LXVI 2015

MODELLING OF MANUFACTURING PROCESSES WITH MEMBRANES

***CRĂCIUNEAN* Daniel Cristian**
SC SoftLine SRL, Sibiu, Romania, daniel@sln.ro

***CRĂCIUNEAN* Vasile**
Faculty of Engineering/Computer & Electrical Engineering Department, "Lucian Blaga" University, Sibiu, Romania, vasile.craciunean@ulbsibiu.ro

Abstract: *The current objectives to increase the standards of quality and efficiency in manufacturing processes can be achieved only through the best combination of inputs, independent of spatial distance between them. This paper proposes modelling production processes based on membrane structures introduced in [4]. Inspired from biochemistry, membrane computation [4] is based on the concept of membrane represented in its formalism by the mathematical concept of multiset. The manufacturing process is the evolution of a super cell system from its initial state according to the given actions of aggregation. In this paper we consider that the atomic production unit of the process is the action. The actions and the resources on which the actions are produced, are distributed in a virtual network of companies working together. The destination of the output resources is specified by corresponding output events.*

Key words: membrane structure, super-cell, action of aggregation, aggregation system of actions with membrane

Introduction

Advanced information technology makes it possible to realize virtual organization in practice, and also the necessities of virtual organizations inspire the information technology [5].

At the Center for Computer Science, Turku, Finland, in November 1998 Gheorghe Păun, proposed a new paradigm of computation: Membrane Computation [3]. This new paradigm of computation seems to be an interesting and original approach to computing inspired from biochemistry. The main new concepts defined are: the membrane, membrane structure and super cell system [3]. More details about these notions are found in [1], [2], [3], [4].

Let X be a set. The family of subsets of X is denoted by $\mathcal{P}(X)$. The cardinality of X is denoted by $|X|$. An alphabet is a finite nonempty set of abstract symbols. For an alphabet V we denote by V^* the set of all strings of symbols in V. The empty string is denoted by λ. The set of nonempty strings over V, that is V^*-{λ}, is denoted by V^+. Let Sub(w) denote the set of subwords of w. Each subset of V^* is called a language over V. The length of a string $x \in V^*$ (the number of symbol occurrences in X) is denoted by $|x|$. The number of occurrences of a given symbol $a \in V$ in $x \in V^*$ is denoted by $|x|_a$. Let Symb(w) denote the set of symbol occurrences in w.

Membrane computation is based on the concept of membrane represented in his formalism by the mathematical concept of multiset. A multiset (over a set X) is a mapping M: $X \rightarrow N \cup \{\infty\}$. For $a \in X$, M(a) is called the multiplicity of a in the multiset M. The support of M is the set supp(M)=$\{a \in X | M(a)>0\}$. A multiset M of finite support, supp(M) = $\{a_1,...,a_n\}$ can be written in the form $\{(a_1, M(a_1)),...,(a_n, M(a_n))\}$. We can also represent this multiset by the string w(M)=$\{a_1^{M(a_1)} ... a_n^{M(a_n)}\}$, as well as by any permutation of w(M).

A membrane structure is a set of labeled multisets with certain restrictions. We define first the language MS over the alphabet {[,]}, whose strings are recurrently defined as follows [1]:
1. $[\,] \in MS$;
2. if $\mu_1,\dots \mu_n \in MS$, $n \geq 1$, then $[\mu_1,\dots \mu_n] \in MS$;
3. nothing else is in MS.

We define now the following relation over the elements of MS: $x \sim y$ if and only if we can write the two strings in the form $x = \mu_1\mu_2\mu_3\mu_4$, $y = \mu_1\mu_3\mu_2\mu_4$, for $\mu_1\mu_4 \in MS$ and $\mu_2, \mu_3 \in MS$. We also denote by $\sim$ the reflexive and transitive closure of the relation $\sim$. This is clearly an equivalence relation. We denote by $\overline{MS}$ the set of equivalence classes of MS with respect to this relation. The elements of $\overline{MS}$ are called membrane structures.

Each matching pair of parentheses [,] appearing in a membrane structure is called a membrane. The number of membranes in a membrane structure μ is called the degree of μ and denoted by deg(μ). The external membrane of a membrane structure μ is called the skin membrane of μ. A membrane which appears in MS in the form [] (no other membrane appears inside the two parentheses) is called an elementary membrane.

The depth of a membrane structure μ, denoted by dep(μ), is defined recurrently as follows [1]:

1. if μ = [ ], then dep(μ) = 1;
2. if $\mu = [\mu_1 \ldots \mu_n] \in MS$, for some $\mu_1, \ldots \mu_n \in MS$ then $dep(\mu) = \max\{dep(\mu_i) \backslash 1 \le i \le n\} + 1$.

2. Aggregation system of actions with membrane

Through the manufacturing process we understand the transformation of a given set of resources (raw materials, semi-finished goods, energy, labor, equipment) into finished products by aggregating specific actions according to their manufacturing recipes.

On the membrane structure Păun adds a set of transforming rules and he therefore obtains the super-cell system [1]. We are introducing the notion of action of aggregation. The manufacturing process is the evolution of a super-cell system from its initial state according to the given actions of aggregation.

In this paper we consider that the atomic production unit of the process is the action [6][7]. The actions and the resources on which the actions are produced, are distributed in a virtual network of companies working together. Actions are very inhomogeneous beginning with actions of design, procurement, logistics, manufacturing, quality control etc. There will be software actions that will store and dynamically process all data on the evolution of the production process. Therefore we begin by defining the notion of action of aggregation on witch our model is based.

Definition 2.1. [8] An action of aggregation *a* is a construct of the form

a = (I,O,M) where:

$I=\{(r_i,c_i,e_i) \mid i=1,2,...m$; r_i is a resource, c_i is the quantity of that resource, e_i is an event};

The resource r_i is called an input resource and represents the necessary resource in the c_i quantity for the action a, and e_i is called an input event.

$O=\{(r_o,c_o,e_o) \mid o=1,2,...n$; r_o is a resource or a final product, c_o is the quantity of the resource, e_o is an event}; If r_o is the resource then it's called the output resource and c_o is the quantity of that resource, else r_o is the final output product. The event e_o is the output event.

M is a set of data and associated metadata such as duration of action, the minimum allowed stock, costs, technical data about the action, software components, etc.

We write below the inputs (I) and outputs (O) of the form (r, c, e) in the form (r^c, e).

Definition 2.2. An aggregation system of actions with membrane of degree m, m≥ 1, is a construct of the form:

$S=(V,T,E, \mu,w_1,\ldots, w_m, (A_1,\rho_1),\ldots ,(A_m,\rho_m))$,

where:

(i) V is an alphabet; its elements are called resources;

(ii) $T \subseteq V$ is the output alphabet, its elements are called final products;

(iii) $E=\{\lambda,\forall$, here, out, in i};

(iv) μ is a membrane structure consisting of m membranes, with the membranes and the regions labelled in a one-to-one manner with 1, 2,. . ., m;

(v) w_i ,$1\le i\le m$, are strings representing multisets over V associated with the regions 1, 2, .. . , m of μ;

(vi) A_i ,$1\le i\le m$, are a finite set of actions of aggregation associated with the regions 1,2,... , m of μ;

ρ_i is a partial order relation over A_i , 1 < i < m, specifying a priority relation among actions of A_i.

The membrane structure μ and $w_1,\ldots, w_m$ of an aggregation system of actions with membrane define a super-cell. The m-tuple $(w_1,\ldots, w_m)$ constitutes the initial configuration of S. In general, any sequence $W^t=(w_1^t,\ldots, w_m^t)$, is called configuration of S at the time t.

The sequence of configurations $W^0,W^1,\ldots,W^t,\ldots$, is obtained as follows:

$W^0=(w_1,\ldots, w_m)$, the initial configuration of S;

$W^{t+1}=(w_1^{t+1},\ldots, w_m^{t+1})$ is the result of applying all possible actions of aggregation on the configuration W^t.

An action of aggregation is executed at time t if it has the highest priority and there are all necessary resources to its execution and other activities in the same region that consumes the same resources. The execution of an action determine appropriate resource consumption.

The destination of the output resources is specified by corresponding output events. The result of using the action is determined by the output event. If a resource appears in O in the form (r^c, here) then it will remain in the same region. If a resource appears in O in the form (r^c, out) then the resource will exit the membrane and will become of the region immediately outside it. In this way, it is possible that a resource leaves the system: if it goes outside the skin of the system, then it never comes back. This is the case of the final products. If an action appears in the form (r^c, in i) , then a will be added to the membrane i, providing that the recurce is adjacent to the membrane i, otherwise the execution of the action is not allowed.

Example 2.1. Consider the aggregation system of actions with membrane of degree 3:

$S=(V,T,E, \mu, w_1, w_2, w_3,(A_1,\rho_1), (A_2,\rho_2),(A_3,\rho_3))$ where:

$V=\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{15}, r_{16}, r_{17}, r_{18}, q_1,q_2,q_4,q_6,q_7,q_9,q_{15}\}$;

$T=\{ r_{16}, r_{17}, r_{18}\}$;

$E=\{\forall,here,out\}$;

$\mu= [_1\ [_2\ ]_2\ [_3\ ]_3\ ]_1$;

$w_1= q_{15}^9$; $w_2= q_1^6 q_2^9 q_4^3$; $w_3= q_6^{12} q_7^{18} q_9^6$;

$A_1=\{ a_{15}, a_{16}, a_{17}, b_{15}, b_{16} \}$; $\rho_1 =\{b_{15}>b_{16}\}$;

a_{15}: I={(r_5^1,∀), (r_{15}^1,∀)}; O={(r_{16}^1,out), (q_{15}^1,here) };

a_{16}: I={(r_{10}^1,∀), (r_{15}^1,∀)}; O={(r_{17}^1,out), (q_{15}^1,here) };

a_{17}: I={(r_{12}^1,∀), (r_{15}^1,∀)}; O={(r_{18}^1,out), (q_{15}^1,here) };

b_{15}: I={(q_{15}^9,∀)}; O={(r_{15}^9,here) };

b_{16}: I={(q_{15}^6,∀)}; O={(r_{15}^6,here) };

$A_2=\{ a_1, a_2, b_1, b_{11}, b_2, b_{21}, b_4, b_{42}\}$; $\rho_2 =\{ b_1>b_{11};\ b_2>b_{21}; b_4>b_{41}\}$;

a_1: I={(r_1^2,∀), (r_2^3,∀)}; O={(r_3^1,here) , (q_1^2,here), (q_2^3,here)};

a_2: I={(r_3^1,∀), (r_4^1,∀)}; O={(r_5^1,out), (q_4^1,here)};

b_1: I={(q_1^6,∀)}; O={(r_1^6,here) };

b_{11}: I={(q_1^4,∀)}; O={(r_1^4,here) };

b_2: I={(q_2^9,∀)}; O={(r_2^9,here) };

b_{21}: I={(q_2^6,∀)}; O={(r_2^6,here) };

b_4: I={(q_4^3,∀)}; O={(r_4^3,here) };

b_{41}: I={(q_4^2,∀)}; O={(r_4^2,here) };

$A_3=\{ a_6, a_7, a_8, b_6, b_{61}, b_7, b_{71}, b_9, b_{91}\}$; $\rho_3 =\{ b_6>b_{61};\ b_7>b_{71}; b_9>b_{91}\}$;

a_6: I={(r_6^4,∀), (r_7^6,∀)}; O={(r_8^1,here), (r_{11}^1,here) , (q_6^4,here), (q_7^6,here)};

a_7: I={(r_8^1,∀), (r_9^1,∀)}; O={(r_{10}^1,out), (q_9^1,here)};

a_8: I={(r_{11}^1,∀), (r_9^1,∀)}; O={(r_{12}^1,out), (q_9^1,here)};

b_6: I={(q_6^{12},∀)}; O={(r_6^{12},here) };

b_{61}: I={(q_6^8,∀)}; O={(r_6^8,here) };

b_7: I={(q_7^{18},∀)}; O={(r_7^{18},here) };

b_{71}: I={(q_7^{12},∀)}; O={(r_7^{12},here) };

b_9: I={(q_9^6,∀)}; O={(r_9^6,here) };

b_{91}: I={(q_9^4,∀)}; O={(r_9^4,here) };

In membrane 2 we have:

initial configuration : $q_1^6 q_2^9 q_4^3$;

They will perform the following actions:

t_1: b_1((q_1^6,∀))→((r_1^6,here));

b_2((q_2^9,∀))→((r_2^9,here));

b_4((q_4^3,∀))→((r_4^3,here));

configuration after t_1: $r_1^6 r_2^9 r_4^3$;

t_2: a_1((r_1^2,∀), (r_2^3,∀)) →((r_3^1,here) , (q_1^2,here), (q_2^3,here));

configuration after t_2: $r_1^4 r_2^6 r_3^1 r_4^3 q_1^2 q_2^3$;

t_3: $a_1((r_1^2,\forall), (r_2^3,\forall)) \rightarrow ((r_3^1,\text{here}), (q_1^2,\text{here}), (q_2^3,\text{here}))$;
$a_2((r_3^1,\forall), (r_4^1,\forall)) \rightarrow ((r_5^1,\text{out}), (q_4^1,\text{here}))$;
configuration after t_3: $r_1^2 r_2^3 r_3^1 r_4^2 q_1^4 q_2^6 q_4^1$;
t_4: $a_1((r_1^2,\forall), (r_2^3,\forall)) \rightarrow ((r_3^1,\text{here}), (q_1^2,\text{here}), (q_2^3,\text{here}))$;
$a_2((r_3^1,\forall), (r_4^1,\forall)) \rightarrow ((r_5^1,\text{out}), (q_4^1,\text{here}))$;
$b_{11}((q_1^4,\forall)) \rightarrow ((r_1^4,\text{here}))$;
$b_{21}((q_2^6,\forall)) \rightarrow ((r_2^6,\text{here}))$;
configuration after t_4: $r_1^4 r_2^6 r_3^1 r_4^1 q_1^2 q_2^3 q_4^2$;
t_5: $a_1((r_1^2,\forall), (r_2^3,\forall)) \rightarrow ((r_3^1,\text{here}), (q_1^2,\text{here}), (q_2^3,\text{here}))$;
$a_2((r_3^1,\forall), (r_4^1,\forall)) \rightarrow ((r_5^1,\text{out}), (q_4^1,\text{here}))$;
$b_{41}((q_4^2,\forall)) \rightarrow ((r_4^2,\text{here}))$;
configuration after t_4: $r_1^4 r_2^6 r_3^1 r_4^1 q_1^2 q_2^3 q_4^2$;
Note that this is identical to the time t3 and so from now on in this membrane process is repeated.
Let's see what resources are sent in the outer membrane:
t_1: λ; t_2: λ; t_3: r_5^1; t_4: r_5^1; t_5: r_5^1;
In membrane 3 we have:
initial configuration : $q_6^{12} q_7^{18} q_9^6$;

They will perform the following actions:
t_1: $b_6((q_6^{12},\forall)) \rightarrow ((r_6^{12},\text{here}))$;
$b_7((q_7^{18},\forall)) \rightarrow ((r_7^{18},\text{here}))$;
$b_9((q_9^6,\forall)) \rightarrow ((r_9^6,\text{here}))$;
configuration after t_1: $r_6^{12} r_7^{18} r_9^6$;
t_2: $a_6((r_6^4,\forall), (r_7^6,\forall)) \rightarrow ((r_8^1,\text{here}), (r_{11}^1,\text{here}), (q_6^4,\text{here}), (q_7^6,\text{here}))$;
configuration after t_2: $r_6^8 r_7^{12} r_8^1 r_9^6 r_{11}^1 q_6^4 q_7^6$;
t_3: $a_6((r_6^4,\forall), (r_7^6,\forall)) \rightarrow ((r_8^1,\text{here}), (r_{11}^1,\text{here}), (q_6^4,\text{here}), (q_7^6,\text{here}))$;
$a_7((r_8^1,\forall), (r_9^1,\forall)) \rightarrow ((r_{10}^1,\text{out}), (q_9^1,\text{here}))$;
$a_8((r_{11}^1,\forall), (r_9^1,\forall)) \rightarrow ((r_{12}^1,\text{out}), (q_9^1,\text{here}))$;
configuration after t_3: $r_6^4 r_7^6 r_8^1 r_9^4 r_{11}^1 q_6^8 q_7^{12} q_9^2$;
t_4: $a_6((r_6^4,\forall), (r_7^6,\forall)) \rightarrow ((r_8^1,\text{here}), (r_{11}^1,\text{here}), (q_6^4,\text{here}), (q_7^6,\text{here}))$;
$a_7((r_8^1,\forall), (r_9^1,\forall)) \rightarrow ((r_{10}^1,\text{out}), (q_9^1,\text{here}))$;
$a_8((r_{11}^1,\forall), (r_9^1,\forall)) \rightarrow ((r_{12}^1,\text{out}), (q_9^1,\text{here}))$;
$b_{61}((q_6^8,\forall)) \rightarrow ((r_6^8,\text{here}))$;
$b_{71}((q_7^{12},\forall)) \rightarrow ((r_7^{12},\text{here}))$;
configuration after t_4: $r_6^8 r_7^{12} r_8^1 r_9^2 r_{11}^1 q_6^4 q_7^6 q_9^4$;
t_5: $a_6((r_6^4,\forall), (r_7^6,\forall)) \rightarrow ((r_8^1,\text{here}), (r_{11}^1,\text{here}), (q_6^4,\text{here}), (q_7^6,\text{here}))$;
$a_7((r_8^1,\forall), (r_9^1,\forall)) \rightarrow ((r_{10}^1,\text{out}), (q_9^1,\text{here}))$;
$a_8((r_{11}^1,\forall), (r_9^1,\forall)) \rightarrow ((r_{12}^1,\text{out}), (q_9^1,\text{here}))$;
$b_{91}((q_9^4,\forall)) \rightarrow ((r_9^4,\text{here}))$;
configuration after t_5: $r_6^4 r_7^6 r_8^1 r_9^4 r_{11}^1 q_6^8 q_7^{12} q_9^2$;
Note that this is identical to the time t3 and so from now on in this membrane process is repeated.
Let's see what resources are sent in the outer membrane:
t_1: λ; t_2: λ; t_3: $r_{10}^1 r_{12}^1$; t_4: $r_{10}^1 r_{12}^1$; t_5: $r_{10}^1 r_{12}^1$;
In membrane 1 we have:
initial configuration : q_{15}^9;

They will perform the following actions:
t_1: $b_{15}((q_{15}^9,\forall)) \rightarrow ((r_{15}^9,\text{here}))$;
configuration after t_1: r_{15}^9;
t_2: No action.
configuration after t_2: r_{15}^9;
t_3: No action.

configuration after t_3: $r_5^1 r_{10}^1 r_{12}^1 r_{15}^9$;
t_4: a_{15} $((r_5^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{16}^1,\text{out}), (q_{15}^1,\text{here}))$;
a_{16} $((r_{10}^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{17}^1,\text{out}), (q_{15}^1,\text{here}))$;
a_{17} $((r_{12}^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{18}^1,\text{out}), (q_{15}^1,\text{here}))$;
configuration after t_4: $r_5^1 r_{10}^1 r_{12}^1 r_{15}^6 q_{15}^3$;
t_5: a_{15} $((r_5^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{16}^1,\text{out}), (q_{15}^1,\text{here}))$;
a_{16} $((r_{10}^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{17}^1,\text{out}), (q_{15}^1,\text{here}))$;
a_{17} $((r_{12}^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{18}^1,\text{out}), (q_{15}^1,\text{here}))$;
configuration after t_5: $r_5^1 r_{10}^1 r_{12}^1 r_{15}^3 q_{15}^6$;
t_6: a_{15} $((r_5^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{16}^1,\text{out}), (q_{15}^1,\text{here}))$;
a_{16} $((r_{10}^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{17}^1,\text{out}), (q_{15}^1,\text{here}))$;
a_{17} $((r_{12}^1,\forall), (r_{15}^1,\forall)) \rightarrow ((r_{18}^1,\text{out}), (q_{15}^1,\text{here}))$;
$b_{16}((q_{15}^6,\forall)) \rightarrow ((r_{15}^6,\text{here}))$;
configuration after t_6: $r_5^1 r_{10}^1 r_{12}^1 r_{15}^6 q_{15}^3$;

Note that this is identical to the time t4 and so from now on the process in this membrane will be repeated.

Let's see what resources are sent out of the system:
t_1: λ; t_2: λ; t_3: λ; t_4: $r_{16}^1 r_{17}^1 r_{18}^1$; t_5: $r_{16}^1 r_{17}^1 r_{18}^1$; t_6: $r_{16}^1 r_{17}^1 r_{18}^1$;

The parallel activity in region i, at time t is denoted by h_i^t and is represented by a word over A_i, which consists of all actions executed in the region at time t. The system parallel activity at time t is a m-tuple $h(t)=(h_1^t,\ldots, h_m^t)$, where each h_i^t is parallel activity in region i at time t.

The evolutionary trajectory of the system S at time t is calculated as follows:

(i) $H^0=(\lambda,\ldots,\lambda)$;

(ii) $H^t=(\prod_{z=0}^{t} h_1^z,\ldots, \prod_{z=0}^{t} h_m^z)$.

Let $W^t=(w_1^t,\ldots, w_m^t)$ the system configuration at time t. We denote by W the set of all possible configurations of the system $W=\{ W^t|t\geq 0\}$. If W is a finite set we say that the system is bounded.

Proposition 2.1. Let S, a bounded aggregation system of actions with membrane and $H^t=(\alpha_1^t,\ldots, \alpha_m^t)$ the evolutionary trajectory of the system at time t. Then there is a z>0 and $\alpha_i,\beta_i, \gamma_i^t \in A_i^*$; $1\leq i\leq m$, so that: $H^t=(\alpha_1(\beta_1)^{\tau(t)}\gamma_1^{\nu(t)},\ldots, \alpha_m(\beta_m)^{\tau(t)}\gamma_m^{\nu(t)})$ for all $t\geq p$.

Proof. Let $W^t=(w_1^t,\ldots, w_m^t)$ the system configuration at time t. Because S is bounded there exists p and q, p<q, so that we have $W^p=W^q$. We believe that p and q are the smallest with this property. Then for $t\geq p$ we have $W^t= W^{t+(q=p)}$. But there is a bijective correspondence between W^t and h(t) and therefore h(t)=h(t+p-q) for all $t\geq p$.

We denote: z=p, $\tau(t)=[(t-p+1)/(q-p)]$, $\nu(t)= (t-p+1) \bmod (q-p)$, and for all $1\leq i\leq m$ we have $\alpha_i=\prod_{z=0}^{p-1} h_i^z$, $\beta_i=\prod_{z=p}^{q} h_i^z$, $\gamma_i^{\nu(t)}=\prod_{z=p}^{\nu(t)} h_i^z$. In these conditions the sentence is verified.

Note that in example 2.1. we have: z=4, $\nu(t)\in\{0,1\}$,
$\alpha_1=b_{15}$, $\alpha_2=b_1b_2b_4a_1a_1a_2$, $\alpha_3=b_6b_7b_9a_6a_6a_7a_8$,
$\beta_1= a_{15}a_{16}a_{17}a_{15}a_{16}a_{17}b_{16}$, $\beta_2= a_1a_2b_{11}b_{21}a_1a_2b_{41}$, $\beta_3= a_6a_7a_8b_{61}b_{71}a_6a_7a_8b_{91}$,
$\gamma_1^0=\lambda$, $\gamma_1^1= a_{15}a_{16}a_{17}$, $\gamma_2^0=\lambda$, $\gamma_2^1= a_1a_2b_{11}b_{21}$, $\gamma_3^0=\lambda$, $\gamma_3^1= a_6a_7a_8b_{61}b_{71}$.

3. Conclusions

The purpose of this model is to demonstrate the usefulness of the membrane and super-cell in the production process modeling by combining independent actions. A process as defined, will evolve into a parallel and distributed environment. Starting the process implies parallel activation of all actions of aggregation.

This model can be developed in similar conditions, by adding new features and additional functionality. Note that, impliedly, we considered that all actions are executed in a unit of time. It is evident that in this model we can take into consideration the introduction of the execution time of each action.

In the model presented in this work, the number of membranes remain constant during evolution. We can consider that the system contains some actions that multiply or dissolve membranes in certain conditions. All this, and more, will be the subjects of other future works.

References

- Calude Ch. S., Păun Gh., Computing with Cells and Atoms an introduction to quantum, DNA and membrane computing, Taylor & Francis ,(2001).
- Dassow J., Păun Gh., On the power of membrane computing, J. Universal Computer Sci., 5, 2, 33-49, (1999).
- Gh. Paun, Computing with membranes, J. Computer System Sciences,2000), in press, and TUCS Research Report No. 208, November, (http://www.tucs.fi),(1998).
- Gh. Paun, Computing with membranes - A variant: P systems with polarized membranes, Intern. J. of Foundations of Computer Science,11, 1 (2000), 167-182, and CDMTCS Research Report No. 098,(ww. cs.auckland.ac.nz/CDMTCS),(1999).
- Mowshowitz A., Virtual Organization: A vision of Management in the Information Age, The Information, (1994).
- Vasile Craciunean, A model of optimal aggregation of actions in manufacturing processes - ACADEMIC JOURNAL OF MANUFACTURING ENGINEERING, VOL. 12, ISSUE 4/2014,p. 30-35,(2014).
- Vasile Craciunean,Generative grammars in actions aggregation- ACADEMIC JOURNAL OF MANUFACTURING ENGINEERING, VOL. 12, ISSUE 4/2014,p.36-41,(2014).
- Vasile Craciunean, A Linguistic Model of Agregation in Virtual Manufacturing Processes - ACADEMIC JOURNAL OF MANUFACTURING ENGINEERING, VOL. 12, ISSUE 3/(2014).