

# A Multiset Rule Based Petri net Algorithm for the Synthesis and Secretary Pathway of Plasma Membrane – Glycoprotein

P. Persis Glory<sup>1</sup> J.D. Emerald<sup>2</sup> N.G. David<sup>3</sup>

<sup>1</sup>Department of Computer Application, Madras Christian College, Tambaram, Chennai

<sup>2</sup>Department of Mathematics, Arignar Anna College for women, Wallajapet.

<sup>3</sup>Department of Mathematics, Madras Christian College, Tambaram, Chennai

## **Abstract:**

*Membrane computing is a branch of Natural computing aiming to abstract computing models from the structure and functioning of the living cell. A comprehensive introduction to membrane computing is meant to offer both computer scientists and non-computer scientists an up-to date overview of the field. In this paper, we consider a uniform way of treating objects and rules in P Systems with the help of Multiset rewriting rules. Here the synthesis and secretary pathway of glycoprotein in epithelial cells of small intestine is considered as an example. A natural and finite link is explored between Petri nets and membrane Computing. A Petri net (PN) algorithm combined with P Systems is implemented for the synthesis and secretary pathway of Glycoprotein. To capture the compartmentalization of P Systems, the Petri net is extended with localities and to show how to adopt the notion of a Petri net process accordingly. The algorithm uses symbolic representations of multisets of rules to efficiently generate all the regions associated with the membrane. In essence, this algorithm is built from transport route sharing a set of places modeling the availability of system resources. The algorithm when simulated shows a significant pathway of safe Petri nets.*

## **Keywords:**

*P systems, Membrane Computing, Multiset rewriting rules, Glycoprotein synthesis, Glycocalyx, PTL net, Petri net algorithm*

## **1. Introduction**

The research area of membrane computing is originated as an attempt to formulate a model of computation, motivated by the structure & functioning of a living cell. It mainly involves the role of membranes in compartmentalization of living cells into “protected reactors”. Initial models were based on a cell-like arrangement of membrane delimiting compartments, where multiset of chemicals (called objects) evolve according to the given evolution rule. These evolution rules were either modeling chemical reactions and had the form of (multiset) rewriting rules or they

were inspired by other biological processes, such as passing objects (symport or antiport fashion) and had the form of communication rules.[1,14] The evolution rules are of two types. The first type is the set of rewriting rules that reside inside the regions (Multisets of chemical reactions). The second type is the set of communication rules that are located on membranes (Passing objects through the membrane). This communication rule governs the exchange of objects through the membrane [9].

The next important step in the development of research in membrane computing is to consider other non-hierarchical arrangement of membranes. While hierarchical cell-like arrangement of membranes correspond to trees, the non-hierarchical tissue-like membrane systems consider arbitrary graphs as underlying structures with membranes placed in the nodes and edges correspond to communication channels. The latest development in this area of research is neural-like membrane systems, which are motivated by spiking neural networks. All classes of computing devices considered in membrane computing are now generically called P Systems. Petri nets [2,15,16] on the other hand are an operational model for concurrent systems directly generalizing state machines by their distributed states and local actions.

Research on application of system started relatively late, but it is now a very active research direction. Application areas include biology, biomedicines, commerce, economics, and computer science. An active and important area of research consists of establishing bridges with other well known models of computations, motivated by computer science, mathematics or biology. Membrane structure is a three-dimensional vesicle which can be geometrically considered as a ball in the Euclidean space. Membrane structure delimits a space, separating inside from outside. The inside space delimited by a membrane serves as a “protected reactor “. In this space reactions take place using molecules. This paper focuses on multiset rewriting rules. The main data structure used in membrane computing is a ‘multiset’ (sometimes called a bag), which is essentially, a set with multiplicities, associated with its elements. Three data structures are used in membrane computing. They are Sets (no multiplicity, no order), Multiset (multiplicity, but no order) and Strings (multiplicity and linear order)

## **2. Basic Membrane Systems**

In this section, we discuss membrane systems and basic variant of P systems, each region contains a multiset of symbol-objects, which correspond to the chemicals swimming in a solution in a cell compartment; these chemicals are considered here as unstructured, that is why we describe them by symbols from a given alphabet. The objects evolve by means of evolution rules, which are also localized, associated with the regions of the membrane structure.

### **2.1 Typical Membrane systems**

Membrane systems or P systems are a computational model inspired by the compartmentalization of living cells and the effect this has on how they function. The biochemical reactions taking place in the compartments of a cell are abstracted to rules specifying which and how many new objects (molecules) can be produced from objects of a certain kind and quantity, possibly involving a transfer to a neighboring compartment. We consider a number of notions which are useful in discussing membrane structure. A membrane  $m$  with no upper neighbor is called a skin membrane. A membrane  $m$  with no lower neighbor is called elementary membrane. The space

outside the skin membrane is called the environment. A region is a space delimited by a membrane [9].

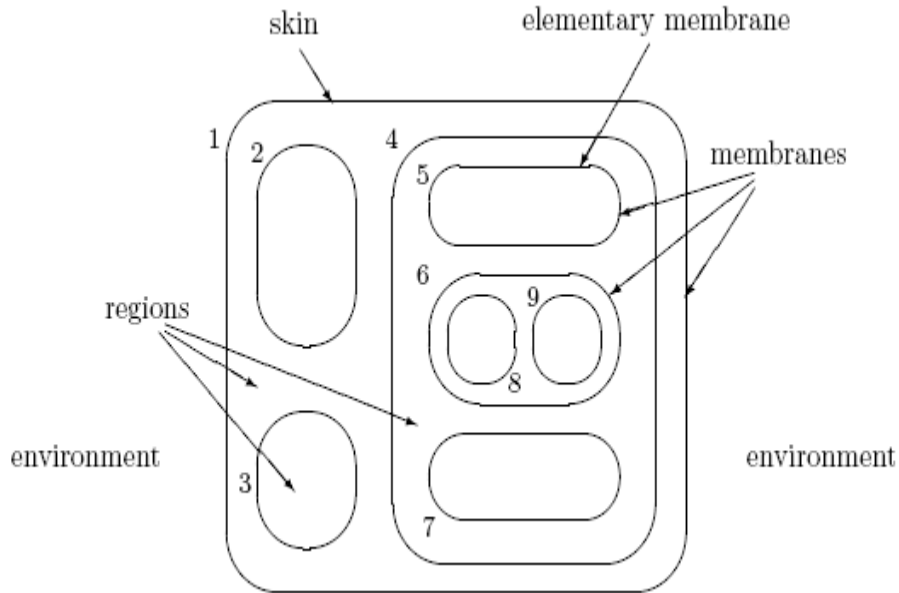


Fig 2.1– A membrane structure

A linear representation of membrane structures, and its standard parentheses notations are discussed. For the above given membrane structure  $S$ , let  $l_1, l_2, \dots, l_n$  be the labels of membranes in  $S$ , where  $l_1$  is the label of the skin membrane. The membrane labeled by  $l_i$ ,  $1 \leq i \leq n$ , is represented by an ordered pair of parentheses  $[ ]_{l_i}$ , and the parentheses expression for  $\mu$  is constructed as follows. A parentheses expression for  $\mu$  is a correct form for  $S$  which includes the right hand parentheses  $]_{l_i}$  for each  $i \in [1, 2, \dots, n]$ . Thus the parentheses expression corresponding to the membrane structure  $S$  in fig 2.2 is

$$[ [ ]_2 [ ]_3 [ ]_4 [ ]_5 [ ]_6 [ ]_7 [ ]_8 [ [ ]_{10} ]_9 ]_1$$

A mathematically natural way to formalize membrane structure is to use rooted un-ordered node-labeled trees. We represent membrane structures through Euler-Venn diagrams because this corresponds directly to the cell-like intuition of compartments, and some crucial notions such as regions come up more naturally in this way.

## 2.2 Synthesis and Transport of Glycoprotein

Glycoprotein (Gp) is a conjugated protein that has covalently bound carbohydrates. The carbohydrate components are oligosaccharides. Glycoproteins (Gp) are abundant on the outer side of the plasma membranes of all the eukaryotic cells, but absent in intracellular organelle's membranes (e.g., mitochondria). In intestinal epithelial cells, microvilli are seen on the apical side of the cells. Each microvillus is surrounded by a loose cell coat called Glycocalyx which is rich in glycoprotein and enzymes. Glycocalyx is a fibrous network of Gp. The Glycocalyx contains hydrolytic enzymes (eg. Sucrose, maltase) that catalyze the final stages in the digestion

of carbohydrates [3,10]. The apical side of the cell is specialized for the absorption of amino acids and glucose from the lumen of the intestine.

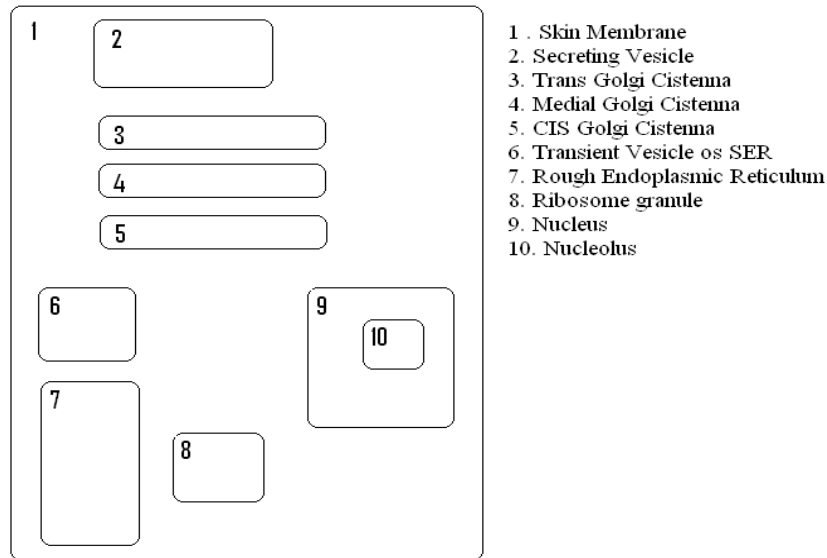


Fig 2.2 – Membrane Structure for the Synthesis and Transport of Glycoprotein

Membrane attached ribosomes, synthesize secretory and membrane proteins. These proteins cross the RER membrane in an unfolded state. Within RER the protein becomes folded. In the lumen of RER the core of Oligosaccharides (carbohydrate) is attached to these folded proteins. Membrane and secretory proteins move via membrane –bounded transport vesicles from the RER to the Golgi complex. The transport route for these proteins in Golgi complex is cis-golgi reticulum – Medial Golgi reticulum – Trans Golgi reticulum. In each region, the side chains of Oligosaccharides are added. Small vesicles transport these proteins from one Golgi region to another Golgi region. The Golgi complex plays a key role in sorting the proteins. The completed and mature glycoprotein formed in Golgi complex is targeted to the secretory vesicles. The secretory vesicles fuse with the plasma membrane and form the cell coat or the Glycocalyx and cover the tip of the microvilli of the epithelial cells. Fig 2.3 shows the complete route for the synthesis of glycoprotein [3, 10].

S.No	Organelle	Multiset rule	In	Out	Target Indications
1	PLASMA MEMBRANE	$r_{11}$	gp	gc	$(y_3, z_5, w, ga_3, na_3, in)$ $(gc, out)$
2	SECRETING VESICLE	$r_{21}$	gp	gp	$(y_3, z_5, w, ga_3, na_3, in)$ $(y_3, z_5, w, ga_3, na_3, out)$
3	TRANS GC	$r_{31}$	e	gp	$(y_3, z_5, w, in)$ $(y_3, z_5, w, ga_3, na_3, out)$
4	MEDIAL GC	$r_{41}$	d	e	$(y_5, z_2, in)$ $(y_3, z_5, w, out)$
5	CIS GC	$r_{51}$	Cp	d	$(y_8, z_2, in)$

					(y <sub>5</sub> ,z <sub>2</sub> , out)
6	SER	r <sub>61</sub>	Cp	Cp	(y <sub>8</sub> ,z <sub>2</sub> , in) (y <sub>8</sub> ,z <sub>2</sub> , out)
7	RER	r <sub>71</sub>	C	Cp	(x <sub>3</sub> , y <sub>9</sub> , z <sub>2</sub> , in) (y <sub>8</sub> ,z <sub>2</sub> , out)
8	RIBOSOME GRANULE	r <sub>81</sub>	b	C	(m, t, u, in) (b, out)
9	NUCLEUS	r <sub>91</sub>	b	b	(b, in) (b, out)
10	NUCLEOLUS	r <sub>101</sub>	a	b	(O <sub>4</sub> , P <sub>70</sub> , in) (b, out)

Fig 2.3 complete transport route of glycoprotein

### 3. Petri Nets and Membrane Computing

In this section, we discuss and study the principles underlying the P systems using symbol objects that happen to be computationally complete. We focus our attention on the processes of these systems and we use Petri nets as tool for our investigations.

#### 3.1 Class of Petri nets

The Petri net is a directed bipartite graph with nodes representing either ‘ places’ (represented by circles) or ‘ transitions ’ (represented by rectangles). A Petri net describes a process in terms of places, transitions and arcs. A Petri net is composed of four parts: a set of places P, a set of transitions T, an input function I and an output function O. The input and output functions relate transitions and places. The input function I is a mapping from a transition to a collection of places, known as input places of transition. The output function O maps a transition to a collection of places known as the output places of the transition. The structure of a Petri net is defined by its places, transitions, input functions and output functions [17].

#### Definition 3.1

A Petri net structure C, is a four-tuple,  $C = (P, T, I, O)$ ,  $P = \{P_1, P_2, \dots, P_n\}$ ,  $n \geq 0$  is a finite set of places,  $T = \{T_1, T_2, \dots, T_m\}$ ,  $m \geq 0$  is a finite set of transition. The set of places and the set of transitions are disjoint,  $P \cap T = \emptyset$ .  $I: T \rightarrow P$  is the input function, a mapping from transitions to bags of places.  $O: T \rightarrow P$  is the output function, a mapping from transitions to bag of places.

#### Definition 3.2

A multiset over a (finite) set X is a function  $m: X \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$  The cardinality of m is  $|m| = \sum_{x \in X} m(x)$ . Multisets over X are represented by strings over X. The set of all (closed) intervals of natural numbers is denoted by INT.

**Definition 3.3**

A membrane structure  $\mu$  (of degree  $m \geq 1$ ) is given by a rooted tree with  $m$  nodes identified with the integers  $1, 2, \dots, m$ . We write  $(i, j) \in \mu$  or  $i = \text{parent}(j)$  to mean that there is an edge from  $i$  (parent) to  $j$  (child) in the tree of  $\mu$ , and  $i \in \mu$  to mean that  $i$  is a node of  $\mu$ .

**Definition 3.4**

Let  $V$  be a finite alphabet of names of objects (molecules). A basic P System over  $\mu$  is a tuple  $\Pi = (V, \mu, \omega_1, \dots, \omega_m, R_1, \dots, R_m)$  such that, for every membrane  $i$ ,  $\omega_i$  is a multiset of objects, and  $R_i$  is a finite set of evolution rules  $r$  of the form  $\text{lhs} \rightarrow \text{rhs}$ , where  $\text{lhs} \neq \lambda$ , is a multiset over  $V$ , and  $\text{rhs}$  is a multiset over

$$V \cup \{ a_{\text{out}} \mid a \in V \} \cup \{ a_{\text{in},j} \mid a \in V \text{ and } (i,j) \in \mu \}$$

such that if  $i$  is the root of  $\mu$ , then no  $a_{\text{out}}$  occurs in  $\text{rhs}$

Petri net is essentially a bipartite directed graph consisting of two kinds of nodes, called places and transitions. Places determine the distribution of the system states over local states, whereas transitions are local actions which when they occur affect only the information provided by adjacent local states. In Place/Transition nets, a typical and prominent Petri net model, system states indicate the local availability of resources by marking each place with number of tokens. In this paper, Multiset calculus is also basic for transforming the token distribution in PT- nets and fig 3.1 shows the initial configuration of P systems, with the rules included. Behavior of PT- nets is captured in a natural way by step semantics in terms of arbitrary combinations (multisets or steps) of concurrently occurring transitions. Like P systems, Petri nets are in essence multiset rewriting systems. [5,7]

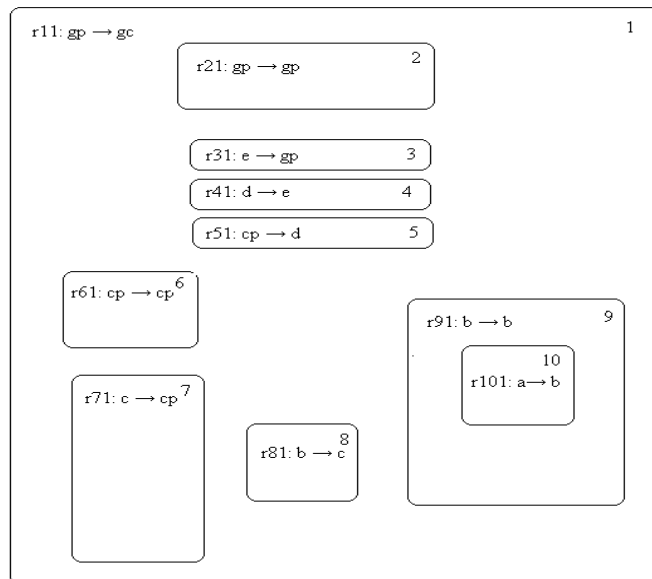


Fig 3.1 – Initial configuration of P systems, rules included.

### 3.2 Place/Transition Nets with localities

We now introduce the class of Petri nets to be used for a direct, behavior preserving translation from basic P systems. To model membrane systems, multisets of places are used to represent the availability of molecules within the compartments, while transitions correspond to evolution rules. As a consequence, each transition is associated with a compartment and this information is explicitly added to the model through the concept of locality of a transition.

To model a basic P system as a PTL-net, we introduce a separate place (a,j) for each molecule a and membrane j . For each rule r associated with a compartment i we introduce a separate transition  $t_i^r$  with locality i. If the transformation described by a rule r of compartment i consumes k copies of molecules a from compartment j , then we introduce a k –weighted arc from place (a,j) to transition  $t_i^r$  , and similarly for molecules being produced [11,12]. Finally, assuming that, initially, compartment j contained n copies of molecule a , we introduce n tokens into place of (a, j) and this idea is formalized as follows

Fig 3.2 shows the Multiset rule based Petri net for synthesis of Glycoprotein with ten different places ( $P_i$ , where  $1 \leq i \leq 10$  ) that represent ten different organelles in the cell namely Nucleolus, Nucleus, Ribosome, Rough Endoplasmic Reticulum, Smooth Endoplasmic Reticulum, Cis Golgi, Medial Golgi, Trans Golgi, Secretary Vesicle , and exocytosis of mature glycoprotein thru membrane. And similarly the transitions ( $t_i$ , where  $1 \leq i \leq 9$  ),represent the chemical reactions that take place in each transitions respectively

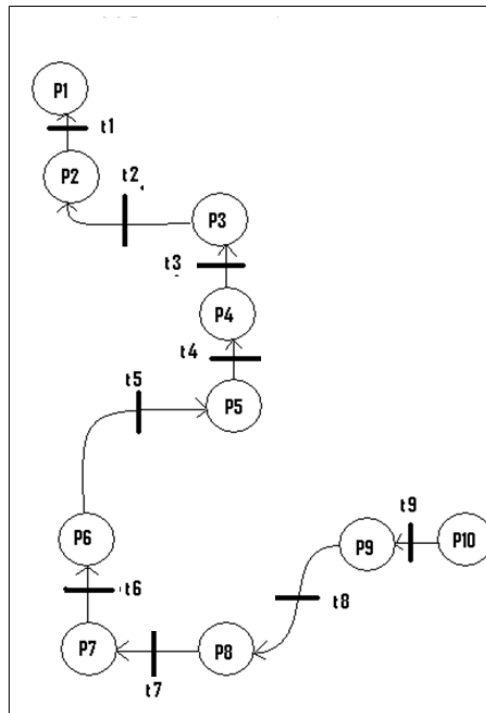


Fig 3.2 Multiset rule based Petri net for synthesis of Glycoprotein

**Definition 3.5**

A PT-Net with localities is a tuple  $NL = (P, T, W, D, M_0)$  where  $P$  and  $T$  are finite disjoint sets of respectively the places and transitions,  $D: T \rightarrow N$  is a locality mapping,  $W: (T \times P) \cup (P \times T) \rightarrow N$  is the weight function, and  $M_0: P \rightarrow N$  is the initial marking. We assume that, for every transition small  $t$ , there is a place  $p$  such that  $\omega(p, t) \neq 0$ . Fig 3.3 shows the translation for the example in fig 2.2, where each place  $(x, i)$  is denoted as  $x: i$  and each transition  $t_i^r$  as  $r$ . Here places are drawn as circles and transitions as boxes.

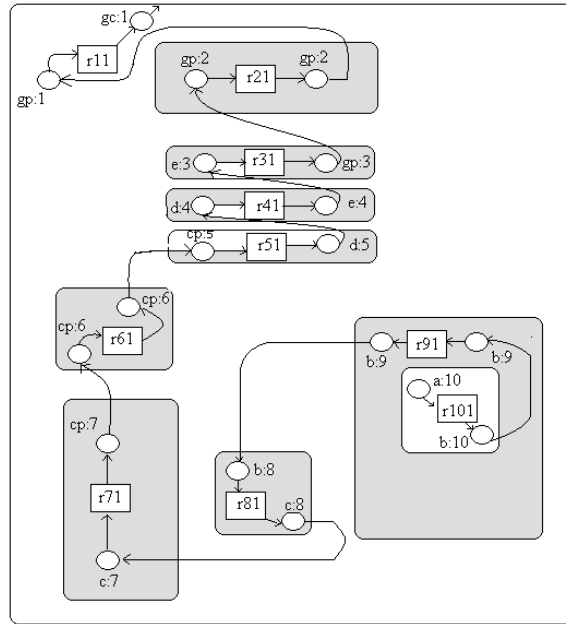


Fig 3.3 PTL – net corresponding to the basic running example

Motivated by natural phenomena for the synthesis and Secretary pathway of Plasma Membrane – Glycoprotein, a whole range of different extensions of the basic membrane system have been introduced in the form of more sophisticated evolution rules. For these extensions, like multiset rewriting rules, the basic translation to PTL-nets can be used [13]. Table 3.1 and 3.2 shows the definition for the Petri net places and transitions



Places	State Definition
P1	Exocytosis of Glycoprotein
P2	Secretary Vesicle
P3	Trans Golgi
P4	Medial Golgi
P5	Cis Golgi
P6	Smooth Endoplasmic Reticulum
P7	Rough Endoplasmic Reticulum
P8	Ribosome
P9	Nucleus
P10	Nucleolous

Table 3.1 : Definition for Places

Transitions	Firing Reaction
t1	r11: gp $\rightarrow$ gc
t2	r21: gp $\rightarrow$ gp
t3	r31: e $\rightarrow$ gp
t4	r41: d $\rightarrow$ e
t5	r51: cp $\rightarrow$ d
t6	r61: cp $\rightarrow$ cp
t7	r71: c $\rightarrow$ cp
t8	r81: b $\rightarrow$ c
t9	r91: a $\rightarrow$ b

Table 3.2 : Definition of Transitions

#### 4. Multiset Rule based Petri net Algorithm

An algorithm is presented for the general Petri net based on a generalization of the basic Multiset rule construction for the synthesis of Glycoprotein which is symmetric with respect to the initial and final marking. Sets of transition sequences described are used for approximations to firing sequences, and the approximation error has to be evaluated during the simulation. The approximation of this algorithm is iterated until a sufficient criterion for reachability can be given, not-withstanding the remaining uncertainty. The algorithm is easy to use and can be easily automated.

```

1: repeat
2:  $t_i = 1$ ;
3: Input ( $P_i$ ); / where  $i = 1 \leq i \leq 10$ 
4: While( $t_i < 10$ ) {
5: for ( $P_i = 10$ ;  $P_i \geq 1$ ;  $P_i--$ ) {
6: for ( $P_j = i-1$ ;  $P_j \geq 1$ ;  $P_j--$ ) {
7:  $t_i = enable$ ;
8:  $P_j = reaction(t_i)$ ; }
9:  $t_i = disable$ ; }
10:  $t_i++$ ; }
11: Until (abnormal condition exists )

```

Fig 4.1 Multiset rule based Petri net Algorithm

## 5. Analysis of the Model

A method to exploit Multiset rule based Petri nets for the synthesis and secretary transport of glycoprotein in the epithelial cells of small intestine is demonstrated. A P Systems strategy and Petri net modeling algorithm are applied to perform the bioprocess functioning and model analysis. With the model, the interrelations between pathway analysis and metabolic chemical are outlined. Diagrammatical results of the dynamics of the chemical transformations and reactions are simulated and observed by developing a simulator in VB.Net and SQL as backend. An explanation of the observed behavior of the transport cycle is proposed in this paper. Finally, the perspective of Petri nets on modeling and simulation of metabolic networks is discussed.

## 6. Concluding Remarks

The present paper should be seen only as a general overview of membrane computing model for the transport of glycoprotein, which is intended to be as pertinent as possible, but, of course, not completely free of a subjective bias. We propose an intriguing relationship between P systems and Petri nets. For a basic model of P systems, this paper presents a new formalization based on (PT- nets), which can adopt one transition to implement the structural and operational semantics of one evolving rule in P systems and to analyze the computation. The translation of basic P systems into PT-nets (with localities) has been carried out in this paper. The main contribution of the work presented is its potential for further development and exploitation of this structural and systematic link between membrane computing and Petri nets. For Petri nets the direct connection with P systems has led to new, interesting notions like locality (PTL –nets). This translation provides a fundamental connection between P systems and Petri nets. It essentially relates evolution rules to Petri net transitions at the most basic level with local aspects of consumption and production of resources taken care of by the correspondence between places and located objects.

Finally the relation between Petri nets and membrane computing as formalized here through the concept of PTL –nets should be compared and combined with fundamental insights and ideas gained in other investigations in the general area of natural computing. The main contribution of the work presented is its potential for further development and exploitation of this structural and systematic link between membrane computing and Petri nets.

## References

- [1] F.Bernardini and M. Gheorghe. On the power of minimal symport/antiport. PreProceedings Workshop on Membrane Computing, WMC-2003, Tarragona, July17-22, 2003.
- [2] J.Desel, W.Reisig, G.Rozenberg,Eds. : Lectures On Concurrency And Petri Nets. Lncs 3098, Springer, 2004.
- [3] E.D.P.De Robertis & E.M.F.De Robertis Jr, Cell & Molecular Biology VIII Edition 248 – 265
- [4] R.Freund and M.Oswald. P systems with activated/prohibited membrane channels. In P̃aun et al. pages 261–269.

- [5] P.Frisco.The conformon-P system: A molecular and cell biology-inspired comutability model. *Theoretical Computer Science*, 312(2-3):295–319, 2004.
- [6] P.Frisco. About P systems with symport/antiport. to appear in *Soft Computing*,2005.
- [7] P.Frisco: P Systems, Petri Nets, and Program Machines, LNCS 3850, Springer, 2006, 209-223.
- [8] P.Frisco and H. J. Hoogeboom. Simulating counter automata by P systems withsymport/antiport. In P`aun et al. pages 288–301.
- [9] Gheorge Paun, Grzegorz Rozenberg & Arto Salomaa, *The Oxford Handbook of Membrane Computing* , Oxfors University Press, 2010, chapters 1,, 2, 3, 5, 15
- [10] Harvey Lodish, David Baltimore et al, *Molecular Cell Biology III Edition* , 669 -735
- [11] H.C.M Kleijn, M.Koutny: Process Semantics Of General Inhibitor Nets. *Information And Computation*, 190(2004), 18-69.
- [12] R.Janicki, P.E.Lauer ,M.Koutny ,R.Devillers: Concurrent And Maximally Concurrent Evolution Of Nonsequentia l Systems. *Theoretical Computer Sci.*,43(1986), 213-238.
- [13] J.Kleijn, M.Koutny: Process Of Petri Nets With Range Testing . *Fundemental Informaticae*, 80(2007), 199-219.
- [14] Mart´ın-Vide, A. P`aun, and G. P`aun. On the power of P systems with symport rules. *The Journal of Universal Computer Science*, 8:317–331, 2002.
- [15] J.L.Peterson : *Petri Net Theory And The Modeling Of Systems*. Prentice Hall,1981.
- [16] W.Reisig, G.Rozenberg, Eds.:*Lectures On Petri Nets . LnCS 1491 And 1492*, Springer,1998.
- [17] Tadao Muratta, *Petri nets : Properties Analysis and Applications*, Proceedings of the IEEE, vol77, No 4 April 1989
- [18] R.Valk: Nets in Computer Organisation. In *Advances in Petri Nets* (W.Brauer, W.Reising, G.Rozenberg, eds.) LNCS 255, Springer, 1986, 218{233.