ABSTRACT

Multiple Sclerosis is an inflammatory demyelinating disorder of the Central Nervous System. It is an autoimmune disease that its causes are still not clarified. The demyelization of the neural axons leads to the physical and cognitive disability of the patient. It is important to model the pathophysiology of Multiple Sclerosis in order to obtain an insight of the problem.

Formal methods help in this direction by providing concepts and disciplines that are applicable to biological systems, too. Such formal methods are Process Algebra, Petri Nets, Automata and Binary Decision Diagrams. In our case study we chose to use Stochastic Petri Nets, considering it to be a consistent, robust and dynamic formalism that can cope with the complexity and diversity of this disease.

Taking as aetiology the recruitment of lymphocytes at inflammatory brain vessels, we modelled the procedure based on data on mice affected by experimental autoimmune encephalomyelitis. The latter is the analogous of Multiple Sclerosis in human beings. We estimated the probability of adhesion and the number of bound molecules as measures to evaluate the acuteness of Multiple Sclerosis. The results were pretty high and actually should be as we are talking about mice suffering from this disease. The use of interferon beta (IFN-β-1b), a medication that reduces the concentration of chemokine, showed some significant reduction in the probability of adhesion and in the number of bound molecules. We ran the simulation for 3-month, 6-month and 12-month therapy. A reduction in adhesion probability of 2.9% is observed by the 12-month therapy compared to the baseline execution. This is quite optimistic for patients suffering from Multiple Sclerosis. Interferon beta is the predominant medication provided to the Multiple Sclerosis patients.

Stochastic Petri Nets formalism is quite easy to learn and use and it is recommended to biologists to use it for their biological simulations. Along with the use of Mobius tool, Stochastic Petri Nets can give powerful perspective to researchers that are interested in non-deterministic phenomena. Biological Systems are such phenomena and should be treated so.

MODELLING THE MULTIPLE SCLEROSIS DISEASE USING STOCHASTIC PETRI NETS

Mikellides Loizos

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science at

University of Cyprus

Recommended for Acceptance

By the Department of Computer Science

January, 2011

APPROVAL PAGE

Master of Science Thesis

MODELLING THE MULTIPLE SCLEROSIS DISEASE USING STOCHASTIC PETRI NETS

Presented by

Mikellides Loizos

Research Supervisor

____

Anna Philippou

Committee Member

____

Costas Pattichis

Committee Member

____

Chryssis Georgiou

University of Cyprus

January, 2011

ACKNOWLEDGEMENTS

Many thanks to my supervisor Dr. Anna Philippou for her guidance in the accomplishment of this thesis. I also thank my family for their full support and patience during this procedure.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

The combination of Computer Science and Biology has formed a new domain called “Systems Biology”. Systems Biology actually combines Biology, Chemistry, Physics, Mathematics, Electrical Engineering and Computer Science. Disciplines from all these fields contribute to this new domain. Computer Science contributes formalisms and simulators to look into the behaviour of biological systems. Different formalisms that were up until now used in modelling software engineering can now model biological systems, too. Formalisms like Process Algebra [1], Petri Nets [2], Automata [3] can help in qualitative or quantitative investigation of systems in Biology. The latter can be either gene regulatory networks, signal transduction networks or metabolic networks. Genes or proteins can be entities in these formalisms that interact, exchange messages or change state. The several simulators that use these formalisms can give in silico experiments that in some cases are more useful than in vivo or in vitro experiments. There are cases, like multiple sclerosis that in vivo experiments are extremely difficult to process and therefore in silico experiments are mandatory.

Biology has now become “executable cell biology” [4] with two types of biological models, the computational and the mathematical. The computational model, given certain data, can mimic biological phenomena, execute and give results that are quite accurate and precise. The mathematical model is handling biological systems as equations with elements interacting and giving products. ODE’s (Ordinary Differential Equations) is a mathematical model that is widely used. It is though a deterministic model to handle natural phenomena. Real life is non-deterministic though. The occurrence of an event is most of the times probabilistic in real life. Two molecules will not certainly interact based on the fact that there is a certain amount of mass present (law of mass). We should better look for the probability of natural phenomena happening. That is where stochasticity is incorporated. Stochastic pi-calculus and Stochastic Petri Nets have a lot to offer in this field.

Our endeavour was to make a good inside view of how biological systems can be modelled with computerized rules and concepts. Literature showed that there are a lot of formal methods that do this: starting from Process algebra, Automata, Binary Decision Diagrams and ending with Petri Nets. We centred our interest on Process algebra and Petri Nets, which are two powerful ways to model concurrent systems. The aim of this study was to understand Systems Biology as systems, including their robustness, design and manipulation. We studied the structure and dynamics of these systems by modelling, simulating and executing the system.

Our case study of examining Multiple Sclerosis using Stochastic Petri Nets showed that this formalism can give good predictions of the possibility to get multiple sclerosis under certain conditions. These conditions were the concentrations of the involved molecules and the rates that characterize their transitions. The recruitment of lymphocytes under inflammation of brain vessels is one of the most common aetiologies of multiple sclerosis. It is considered an autoimmune disease that still has not been cured. Our further step was to examine the effect of interferon beta, IFN-β-1b, on our model and how much is the probability of adhesion reduced due to this change. Interferon beta is a medication given in the cases of multiple sclerosis. The results are quite optimistic on the number of bound molecules but also on the probability of adhesion. Both showed significant reduction.

Computational Systems Biology tries to establish methods and techniques that enable us to understand such systems as systems, including their robustness, design and manipulation. It means to understand: the structures and the dynamics of systems, methods to control, design and modify systems to cope with desired properties. The modelling contributes in a major way to reach these aims by introducing methods for understanding, simulating and predicting the behaviour of the systems [5].

1.2 Thesis Structure

In the next chapters we will describe the Biological Systems in Chapter 2, the Formal Methods in chapter 3, the Petri Nets in Chapter 4, the Case Study in Chapter 5 and finally the Conclusions and future work in Chapter 6.

Chapter 2

Biological Systems

The achievement in recent years to explore the inside of cell and living organisms has led to the study of Biology not only as chemical reactions but as a system where biological entities can have a state, being able to transit to another state. This kind of study has given rise to a new field called “Systems Biology”. The experiments nowadays are not only in vivo or in vitro, but also in silico. Systems Biology can help in the modelling of a biological system and examine its behaviour under certain conditions. The in silico prediction of such a behaviour can evolve useful knowledge for the therapy of diseases and the usage of certain pharmaceutical treatment.

The procedure by which a natural system is converted into a digital system is shown in Figure 1. The measurement and observation of a natural system gives us a biological phenomenon. This leads to a hypothesis as how the natural system works [6]. Induction and modelling give us the chance to prove whether this hypothesis is true. The formal system will give us some results that can be analyzed and explain the natural system. The formal system is usually a simulation, the results of which may vary from the experimental ones. This means that the modelling of the system is wrong and changes should be applied. If the results of the simulation are equivalent to the experimental, then the model is correct. Model checking is usually applied to biological systems to examine the correctness of modelling.

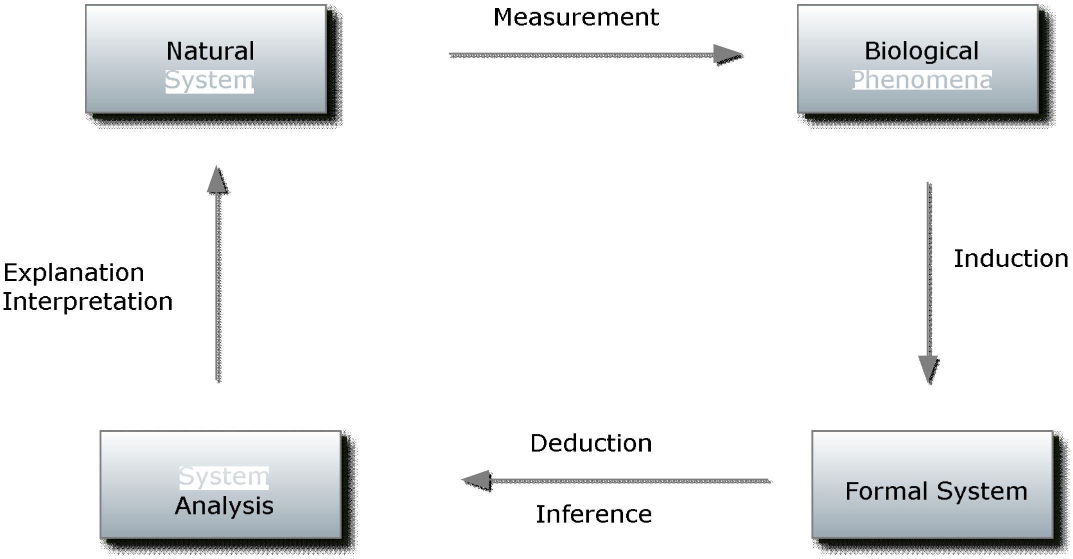


Figure 1: Schematic view of Systems Biology [6]

2.1 Gene Regulatory Networks

Gene Regulatory Networks are one of the most common biological systems that are examined. Considering the fact that each gene expresses one protein, protein can affect the transcription of another gene and so the expression of another protein. The same concept can be applied to another gene, forming in this way a network where each gene can be affected positively or negatively one or more other genes. Such network can be modelled with a graph where a node is the gene and the arcs (activative or inhibitory) denote the positive or negative act on the other gene (Fig. 2). The activative action of in the network is denoted $X \to Y$, whereas the inhibitive action, like this X⊣Y. X and Y are consecutive genes.

.

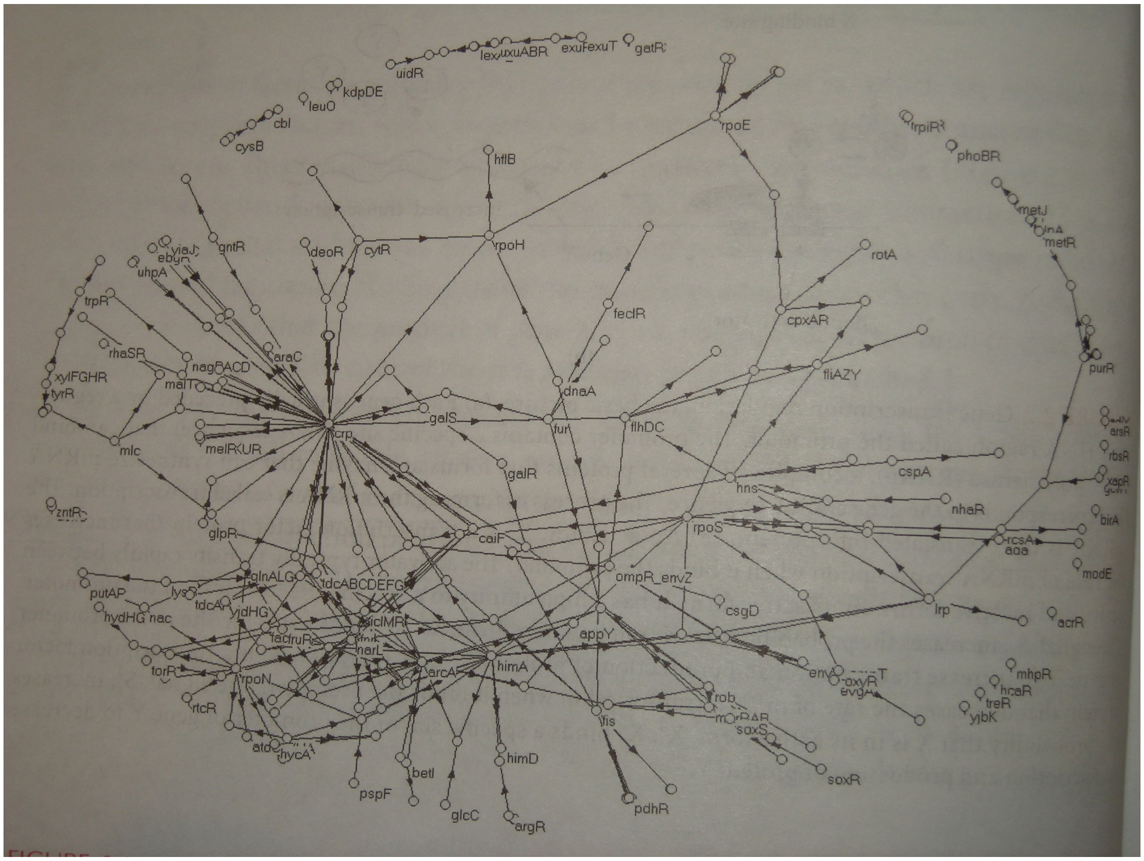


Figure 2: A transcription network that represents about 20% of the transcription interactions in the bacterium E. coli. Nodes are genes (or groups of genes coded on the same mRNA called operons) [7].

2.2 Signal Transduction Networks

Signal transduction is the process by which a cell corresponds to a change in the outer environment. Molecules that are attached to the receptors of the surface result the production of proteins in the inner of the cell. This is a chain reaction, by which protein activates the production of another protein or chemical substance. This leads to the formation of a network - signal transduction networks – that are interesting to model. The whole procedure is vital to the survival of the cell and Systems Biology is interested in this field. Changes in the environment might lead to change of the shape of the cell. Other changes might be changes in temperature or pH.

2.3 Biological Systems vs. Computer Systems

Biological systems have a lot in common and some differences with the computer systems. Biological Systems have, most of the times, larger number of processes with identical behaviour than computer systems. Such processes in biological systems could be thousands of proteins of the same type [8]. However, both systems have common properties like the concurrency. Biological Systems have a lot of processes running concurrently. Proteins bind on the promoter of DNA to promote or inhibit the translation of DNA. This can happen by various proteins at the same time. The stochasticity of this to happen is another property that is frequent in biological systems. Stochasticity can be observed in Computer Systems, too. The communication among computer systems is very common. Nowadays computer networks are capable of sending messages and exchanging information. Biological systems can do so, too. The exchange of molecules during biochemical reactions and the involvement of enzymes in a chemical reaction is a way of communication that can be modelled and examined.

Chapter 3

Formal Methods

In this section we give a description of the formal methods that have been proposed in the literature for modelling Biological Systems. We present Binary Decision Diagrams, Cellular Automata, Petri Nets and Process Algebra, and we compare their relative capabilities.

3.1 BDD’s – MDD’s

Binary decision diagrams (BDDs) are another way of representing Boolean functions. Their simple form is the binary decision trees (Fig. 3) where the non-terminal nodes represent variables x, y, z… and the leaves are labelled with 0’s and 1’s. The non-terminal nodes have one dashed line and one solid one coming out of them. If the variable’s value is 0 the tree follows the dashed line, if the variable’s value is 1 the tree follows the solid line. Reaching the leaves, the label of the leaf is the value of the function [9].

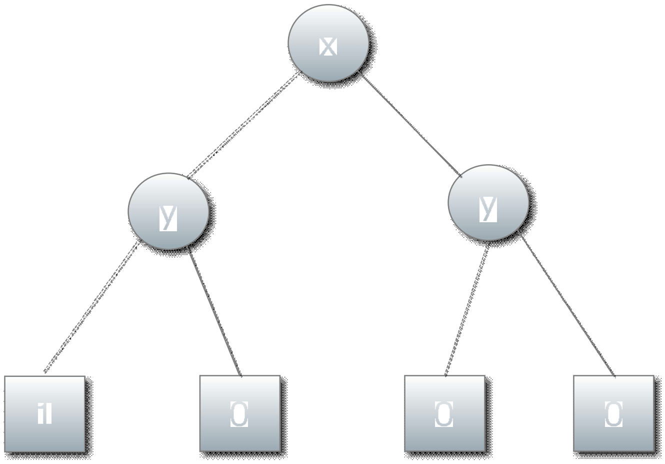
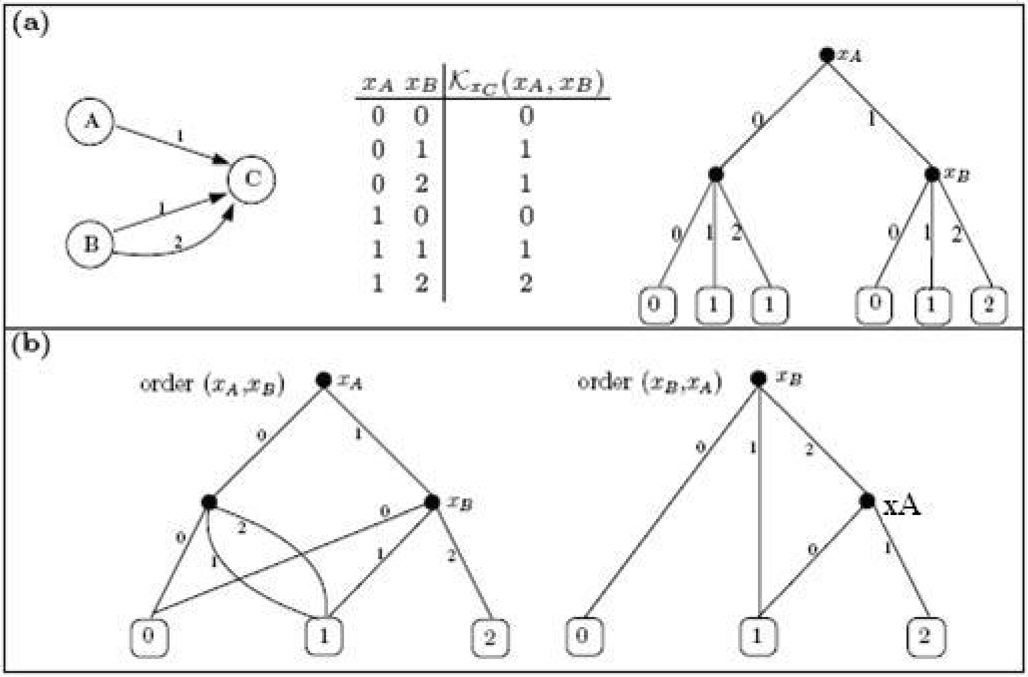


Figure 3: An example of a binary decision tree [9]

For example (Fig.3) the function f(0, 1) has a value of 0, as x=0 and the x node takes the dashed line. Then y =1 and takes the solid line.

Multi-valued decision diagrams (MDD's) (Fig.4) are a generalization of the BDD's where decision nodes have many children, as many as the number of the possible values of the final leaves; which denotes the value of the function [10].



x_A	x_B	$K_{x_C}(x_A, x_B)$
0	0	0
0	1	1
0	2	1
1	0	0
1	1	1
1	2	2

Figure 4: Example of a simple logical regulatory graph with its MDD representation: (a) The regulatory graph, the table defining the function *KC* together with its decision tree representation. (b) The reduced MDDs (considering the two different ordering of *xA* and *xB*) representing *KC*, with *xA* and *xB* as decision variables and the values of *KC* labelling the leaves [10].

BDD's can be used in modelling gene regulatory networks presenting the genes in the form of 0's and 1's. The 0 represents inactive gene whereas 1 represents active gene. The purpose of such formalism is to find stable states of the network. This means states at which there is a stable expression of genes. Regulatory models can be also presented with MDD’s where nodes take multi-values. Stable states in this case can give concentrations of the genes in a range of values (e.g.0-2). BDD’s and MDD’s may be good in qualitative analysis of a network but are not that good for quantitative analysis. In cases where one is looking for the actual concentration of a node, BDD’s and MDD’s are not appropriate for.

3.2 Cellular automata (CA)

Cellular automata (CA) are simple computer simulation tools that can be used to model both temporal and spatiotemporal processes using discrete time and/or spatial steps. Similar to Petri nets, CA models provide a relatively nonmathematical alternative to differential equations for spatiotemporal simulation. CAs normally consist of large numbers of near identical components with local interactions layered on a lattice or grid. The states or values of the components evolve synchronously in discrete time steps according to a set of rules. The value of a particular site is determined by the previous values or the states of the neighbouring sites. Cellular automata were invented in the late 1940s by von Neumann and Ulam and have been used to model a wide range of physical processes, including heat flow, spin networks and reaction–diffusion processes. Cellular automata also have a long history in biological modelling. Indeed, one of the first computer applications in Biology was a CA simulation called Conway’s Game of Life. This simple model simulated the birth, death and interaction between cells randomly placed on a square lattice or grid [3].

3.3 Petri Nets

Petri Nets (PN’s) have been for long used in Software Engineering in modelling computer systems. Now this formalism can also be used in Biology. Their configuration is quite simple and can be applied easily. The process is denoted by a place, a circle, where tokens take place. Tokens can be considered as dots inside the circle. In the case of biological systems they can be the concentration of a molecule species. The number of tokens in one place is the marking of the place. The other element is transition that is designed as rectangle. Transitions are responsible for the movement of tokens from the input places to the output places. Transitions are fired based on a Boolean predicate that has to be true. Once fired, the transition moves tokens from the input place to the output place. Arcs give the flow of the tokens and have a weight on that in the simple form of PN's the marking of the place should be greater or equal to the weight to have a flow of tokens.

Petri Nets can model systems in various ways. Their extensions can provide capabilities that make PN's a powerful tool to model systems. Functional PN's have the capability to place a formula in the place of the weight; a fact that makes PN's more dynamic. While standard PN's have discrete places and transitions, Hybrid PN's have both discrete and continuous places and transitions. This extension makes PN's applicable to systems that have continuous concentrations, i.e. concentrations of real numbers and not integers. We have also in this case the introduction of inhibitory and test arcs. The combination of the two, hybrid and functional, makes a new extension, the Hybrid Functional PN. Finally, we have Stochastic PN's, where the transitions can have time delays and a probabilistic distribution. This takes into consideration the probability of an event to happen and the delay it takes. In the case of modelling, it is the probability of a transition to fire and the delay for the execution of the transaction.

3.3.1 Example

In Figure 5 we see a Petri Net in a simple form. p1, p2 and p3 are the places. m1, m2 and m3 are the markings. 1, 3, 2 on the arcs are the weights. t1 is the transition.

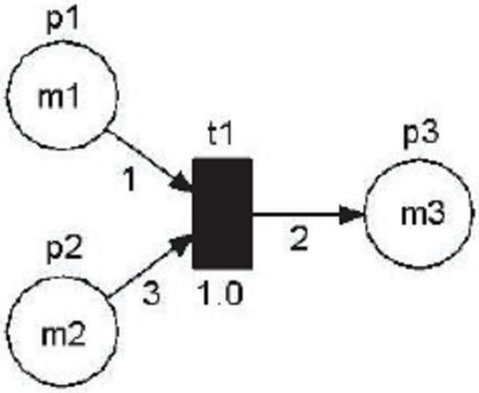


Figure 5: A simple form of Petri Nets.[2]

3.3.2 Related work

Literature has many examples of work done in modelling Biological Systems with Petri Nets. In this section we provide three works that model the glycolitic pathway, the apoptosis and the nutritional stress of E. Coli.

3.3.2.1 Glycolitic Pathway

H. Matsuno et al. [11] in their work have modelled the lac operon gene regulatory mechanism using Hybrid Functional Petri Nets (HFPN) (Fig.6). The glycolitic pathway was also modelled and their purpose was to show the changes of the expression of genes LacZ and LacY and their correspondent proteins as the concentration of lactose decreases. When lactose decreases, the mechanism of glycolitic pathway is activated with the rise of Lac Z and Lac Y. However the mutant lac Z- and lac Y- repress the expression to the two genes respectively and their expression in this case is minimized.

In this case the use of Hybrid Functional Petri Nets is useful as there is a need to model both discrete and continuous places and transitions. There is also the use of inhibitory arcs and test arcs that are available in the Hybrid form.

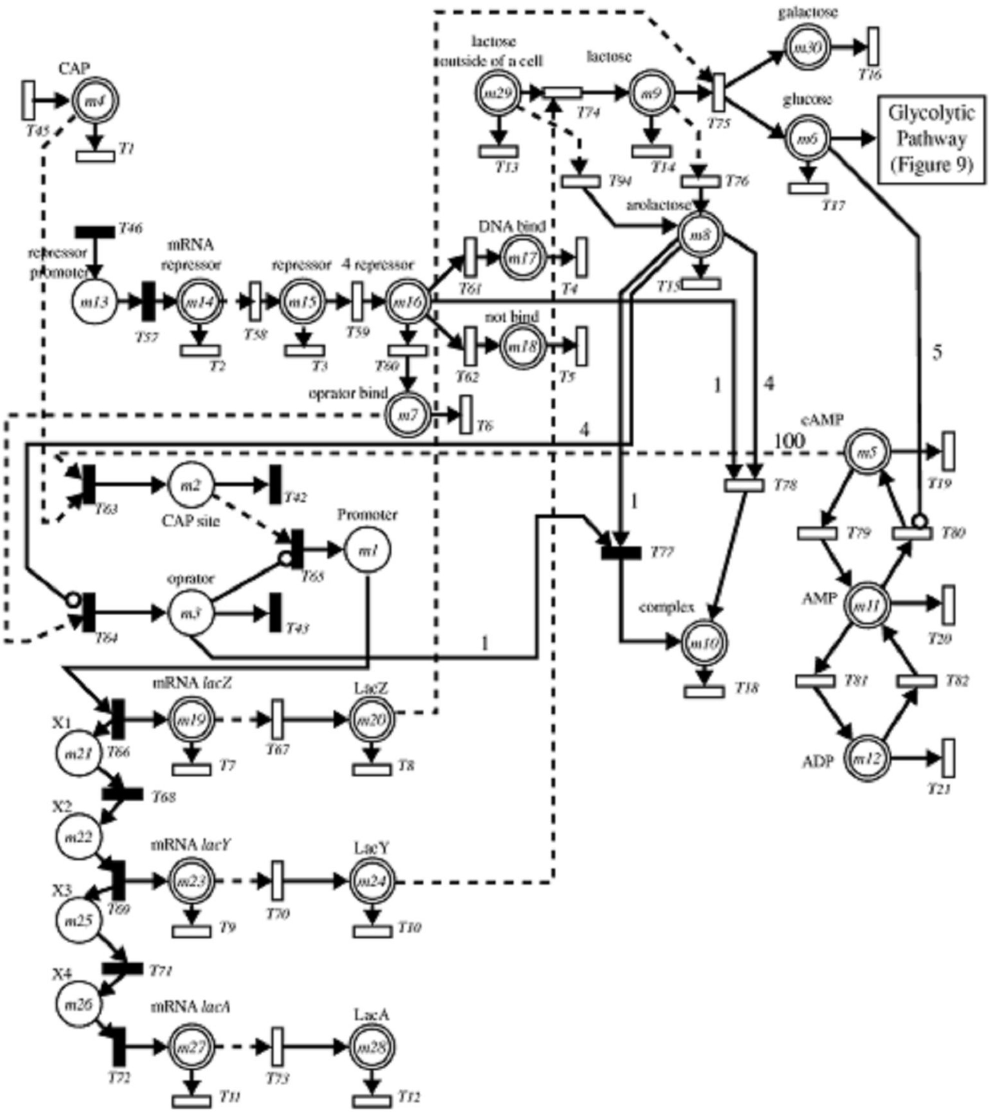


Figure 6: HFPN modelling of the *lac* operon gene regulatory mechanism [11]

3.3.2.2 Apoptosis

Apoptosis is the regulated cell suicide program [12] that is very important to the cell's life cycle. Tissues regulate the number of cells in order to protect themselves from malicious cells or viruses. Several neurological diseases like Alzheimer's or Parkinson's disease are caused due to the disturbance of this life cycle.

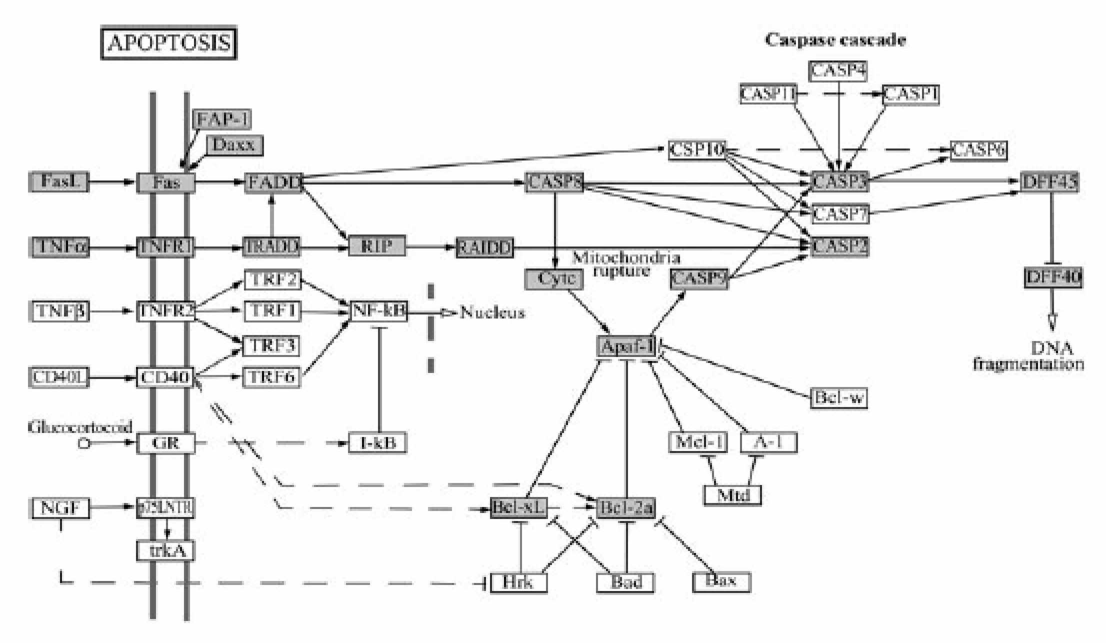


Figure 7: The KEGG representation of apoptosis. Crossbar arrowheads indicate inhibition. Branching arcs go to alternative as well as to concurrent successors. The fragment considered here is highlighted in grey [12].

The work of M. Heiner and I. Koch tried to model this behaviour. Figure 7 shows the KEGG representation of the procedure. KEGG is the Kyoto Encyclopaedia of Genes and Genomes, a series of databases developed the last 10 years involving genome sequences and chemical information. KEGG focuses on the coverage of yeast, mouse and human metabolic pathways [12]. Their modelling is done using Petri Nets.

Their work examines also the carbon metabolism in potato tuber also modelled in Petri Nets. The validation of the model is done with the METATOOL software package. They check the P-invariants and the T-invariants for suitable biological interpretations.

3.3.2.3 Nutritional Stress of E. Coli

A work done by Steggles L.J. et al. [13] is examining the nutritional stress response in E. coli. The system modelled is shown in Figure 8. They used the PEP tool for validating and analyzing the model. PEP is a tool supporting Petri Nets. The initiative for the activation of the system is the *Signal*. This denotes the case where there is presence or absence of carbon starvation. The level of *Stable RNA (SRNA)* is a sign of whether the system is in the exponential phase i.e. high level or in its stationary phase where the system works only for the maintenance of the principal functions

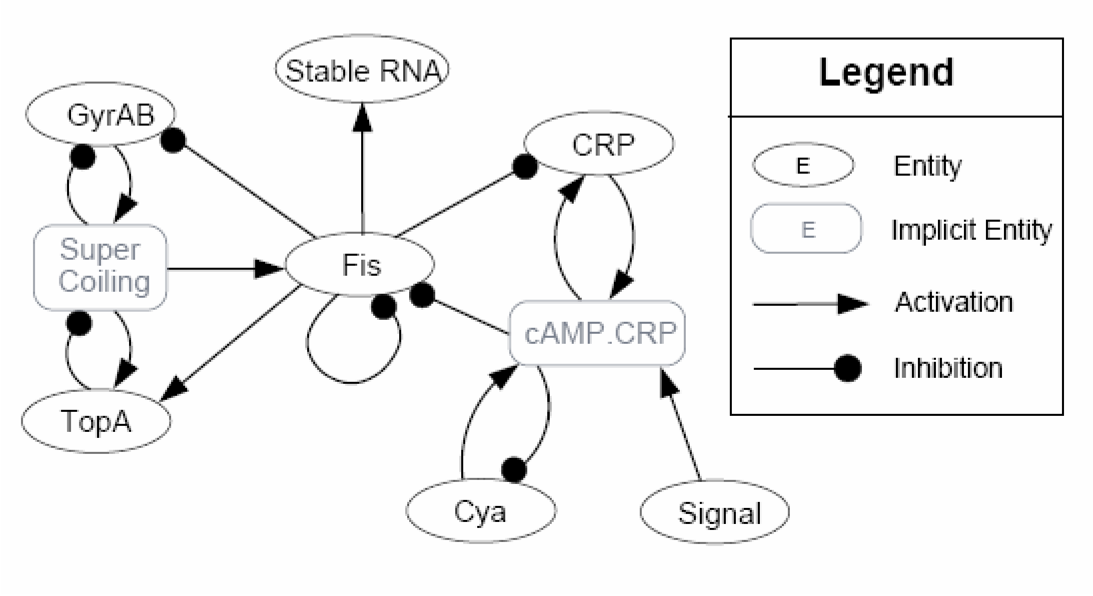


Figure 8: Nutritional stress of E.coli [13].

The system can be expressed in truth tables for each of the entity involved. Using Petri Nets they initialize the system in its exponential phase by assigning the 1 value for *Signal* in all cases. After two periods of time the system switches to the stable state, where *SRNA* = 0. PEP tool helps in the model checking of the system as rules like:

$$GyrAB + TopA > 1,\ \ GyrAB\_Done = 1$$

can be applied as constrains to the system. This is to show that *GyrAB* and *TopA* should be mutually exclusive and *GyrAB_Done* = 1 is to ensure that they consider only states reached after a complete pass of the two-phase commit protocol. PEP in this case is able to confirm that no state satisfying the above rule can be reached from reasonable initial state, so *GyrAB* and *TopA* must be mutually exclusive. In the same way *CRP* and *Fis* can also be shown as mutually exclusive.

3.4 Process Algebra

3.4.1 Definition

Process algebra (process calculus) [1] is a diverse family of related approaches to formally modelling concurrent systems. It provides a tool for the high-level description of interactions, communications, and synchronizations between a collection of independent agents or processes. They also provide algebraic laws that allow process descriptions to be manipulated and analyzed, and permit formal reasoning about equivalences between processes (e.g., using bisimulation). Leading examples of process algebra include CSP, CCS, ACP, and LOTOS. More recent additions to the family include the π-calculus, the ambient calculus, PEPA and the fusion calculus [1].

While the variety of existing process algebras is very large (including variants that incorporate stochastic behavior, timing information, and specializations for studying molecular interactions), there are several features that all process algebras have in common:

- Representing interactions between independent processes as communication (message-passing), rather than as the modification of shared variables
- Describing processes and systems using a small collection of primitives, and operators for combining those primitives

- Defining algebraic laws for the process operators, which allow process expressions to be manipulated using equational reasoning [1].

To define a process calculus, one starts with a set of *names* (or *channels*) whose purpose is to provide means of communication. In many implementations, channels have rich internal structure to improve efficiency, but this is abstracted away in most theoretic models. In addition to names, one needs a means to form new processes from old. The basic operators, always present in some form or other, allow:

- parallel composition of processes
- specification of which channels to use for sending and receiving data
- sequentialization of interactions
- hiding of interaction points
- recursion or process replication [1].

3.4.2 Example

The syntax of agents [14] may be summarized as follows:

$$
\begin{array}{rcll}
P & ::= & \mathbf{0} & (1) \\
 & | & P_1 + P_2 & (2) \\
 & | & \overline{y}x.P & (3) \\
 & | & y(x).P & (4) \\
 & | & \tau.P & (5) \\
 & | & P_1 \mid P_2 & (6) \\
 & | & (x)P & (7) \\
 & | & [x = y]P & (8) \\
 & | & A(y_1, \ldots, y_n) & (9)
\end{array}
$$

Equation 1 shows that the process P is over.

Equation 2 shows that the process P is converted to P1 or to P2.

Equation 3 shows an output of y via channel x.

Equation 4 shows an input of y via channel x.

Equation 5 shows a silent prefix.

Equation 6 shows a parallel execution of P1 and P2.

Equation 7 shows a restriction of process P to accept actions at port x.

Equation 8 shows a match of names x and y.

Equation 9 is a definition of an agent.

3.4.3 Related work

Van Bakel et al. [15] using the stochastic BioAmbients have modelled the endocytotic pathway of Fibroblast Growth Factor (FGF) as shown in Figure 9. Endocytosis is a common communication mechanism in eukaryote cells. It is a mechanism by which the cell membrane envaginates to form a membrane limited vesicle. Vesicles relocate in different compartments inside the cell. Eukaryotic cells continually engage endocytosis to supply the cell with nutrients. There are different causes to endocytosis, however, if initiated by external proteins binding to receptors located on the cell, we speak of *receptor meditated endocytosis*. The extra-cellular protein that initiates the endocytosis is called a *ligand*. The route taken by the vesicle in the receptor meditated endocytosis is well documented in the literature. The vesicle containing the complex ligand-receptor moves to the *sorting endosome* and then to the *late endosome*. At this point the fate of receptors varies: either they are degraded into the *lysosome*, or they reach the membrane via the *recycling endosome*. Receptors are inactive unbound, yet the binding with the ligand activates a chemical signal which in turn could be considered the cause of cell's activity such as stimulation to divide, to migrate or to differentiate into a different cell type. Over-stimulation of such signal is deemed to be responsible for several diseases such as cancer [15]. The modelling of the procedure is shown in Table 1.

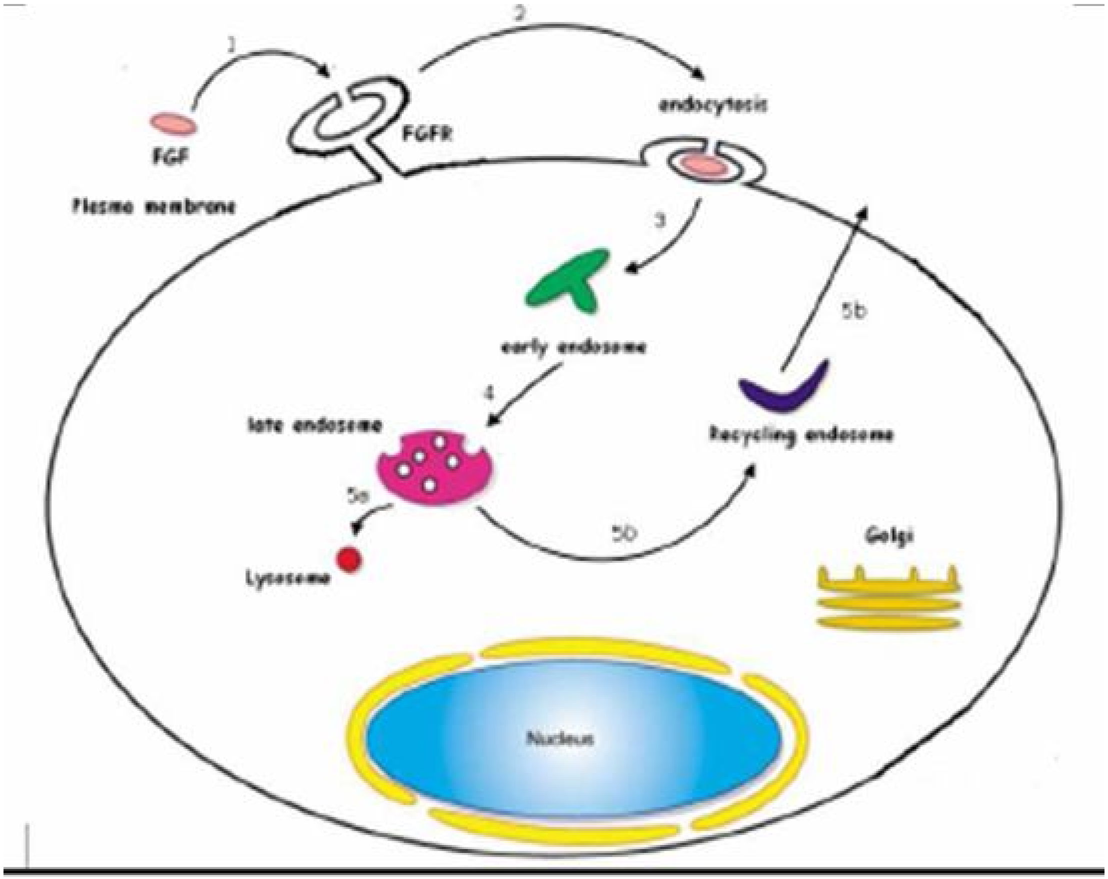
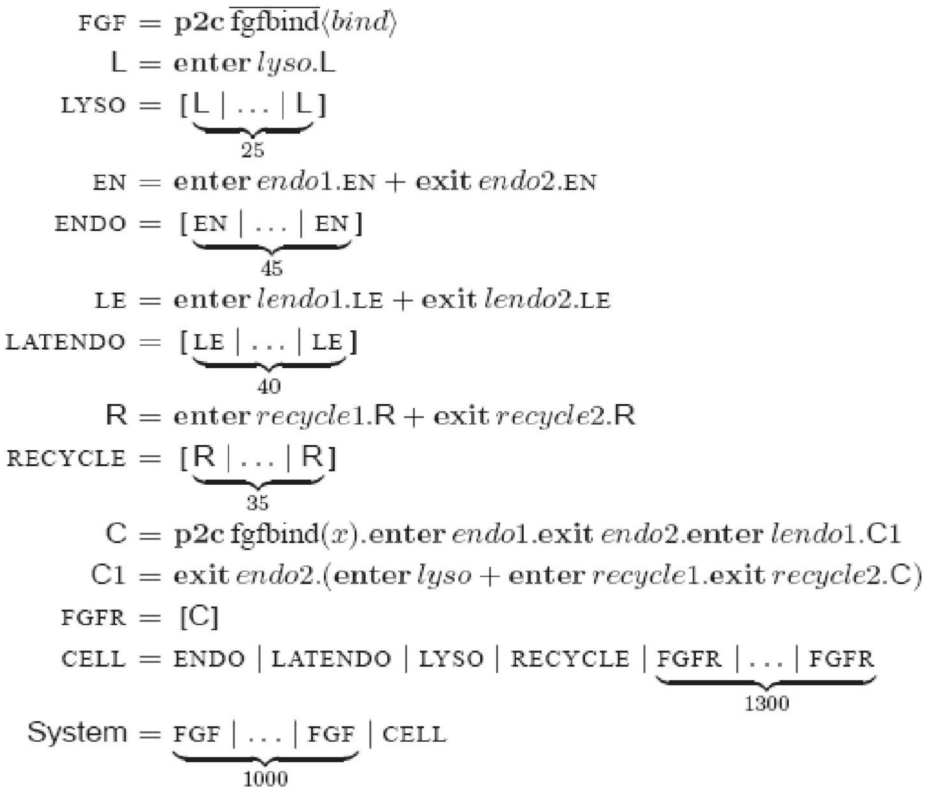


Figure 9: Endocytosis route [15].

Table 1: The implementation of FGF pathway in process algebra (BioAmbient) [15].



$$\begin{aligned}
\text{FGF} &= \mathbf{p2c}\,\overline{\mathbf{fgfbind}}\langle bind\rangle \\
\mathsf{L} &= \mathbf{enter}\,lyso.\mathsf{L} \\
\text{LYSO} &= [\underbrace{\mathsf{L} \mid \ldots \mid \mathsf{L}}_{25}] \\
\text{EN} &= \mathbf{enter}\,endo1.\text{EN} + \mathbf{exit}\,endo2.\text{EN} \\
\text{ENDO} &= [\underbrace{\text{EN} \mid \ldots \mid \text{EN}}_{45}] \\
\text{LE} &= \mathbf{enter}\,lendo1.\text{LE} + \mathbf{exit}\,lendo2.\text{LE} \\
\text{LATENDO} &= [\underbrace{\text{LE} \mid \ldots \mid \text{LE}}_{40}] \\
\mathsf{R} &= \mathbf{enter}\,recycle1.\mathsf{R} + \mathbf{exit}\,recycle2.\mathsf{R} \\
\text{RECYCLE} &= [\underbrace{\mathsf{R} \mid \ldots \mid \mathsf{R}}_{35}] \\
\mathsf{C} &= \mathbf{p2c}\,\mathbf{fgfbind}(x).\mathbf{enter}\,endo1.\mathbf{exit}\,endo2.\mathbf{enter}\,lendo1.\mathsf{C1} \\
\mathsf{C1} &= \mathbf{exit}\,endo2.(\mathbf{enter}\,lyso + \mathbf{enter}\,recycle1.\mathbf{exit}\,recycle2.\mathsf{C}) \\
\text{FGFR} &= [\mathsf{C}] \\
\text{CELL} &= \text{ENDO} \mid \text{LATENDO} \mid \text{LYSO} \mid \text{RECYCLE} \mid \underbrace{\text{FGFR} \mid \ldots \mid \text{FGFR}}_{1300} \\
\mathsf{System} &= \underbrace{\text{FGF} \mid \ldots \mid \text{FGF}}_{1000} \mid \text{CELL}
\end{aligned}$$

3.4.4 Process-algebraic Variations

There are variations of Process Algebra that have been built into tools. Examples are SPiM and BioSPI.

3.4.4.1 SPiM

The Stochastic Pi Machine (SPiM) is a programming language for designing and simulating computer models of biological processes. The language is based on pi-calculus, and the simulation algorithm is based on standard kinetic theory of physical chemistry. The language features a simple graphical notation for modelling a range of biological systems, and can be used to model large systems incrementally, by directly composing simpler models of subsystems. The project is under the Microsoft research group.

3.4.4.2 BioSPI

The BioSPI project is a variation of process algebra modelling biological processes and pathways using the pi-calculus and ambient calculus. Having extensions like stochastic pi-calculus, BioSPI can show the suitability of process algebra for modelling signal transduction networks, metabolic pathways and transcriptional regulation. The benefits of this tool are the formal representation of complex networks, the simulation and monitoring of their behaviour, the formal verification of their properties and the comparison of networks across organisms.

BioSPI is also a computer application, based on Logix system, implementing Flat Concurrent Prolog (FCP). The latter provides mobility and synchronized communication, features that are vital in pi-calculus. BioSPI works on Linux operating system.

3.5 Evaluation

Process algebra is an equational way of dealing with Biological Systems. Petri nets, on the other hand, is a formalism that gives visual representation of the system and can be more understandable by the people. Modelling biological complex systems in both Petri Nets and process algebra presents several advantages. Models can be compositionally built, offering the opportunity to compose parts of the model that are developed at different times by different people; models can be easily manipulated by simply changing some components and evaluating the impact of those changes over the behaviour of the whole model. In silico experiments can be repeated -i.e. several runs of the same model can be performed- with different parameters allowing a simple and effective sensitivity analysis [15].

3.6 Other work

There are several other works done on the field of Systems Biology using tools like PRISM or Live Sequence Charts. We provide a brief overview of two works using these tools.

3.6.1 PRISM Modelling MAPK Cascade Pathway

M. Kwiatkowska et al. in their work [16] have modelled the MAPK cascade pathway, (Fig. 10) which is the vital signal pathway in the growth, proliferation and survival of many cells. MAPKKK being the Mitogen-Activated Protein Kinase Kinase Kinase through several phosphorylations activates MAPK. Using the PRISM tool, they estimate the expected activated MAPK having 4 (N=4) initial quantities of MAPKKK, MAPKK and MAPK or having 8 (N=8) initial quantities of them.

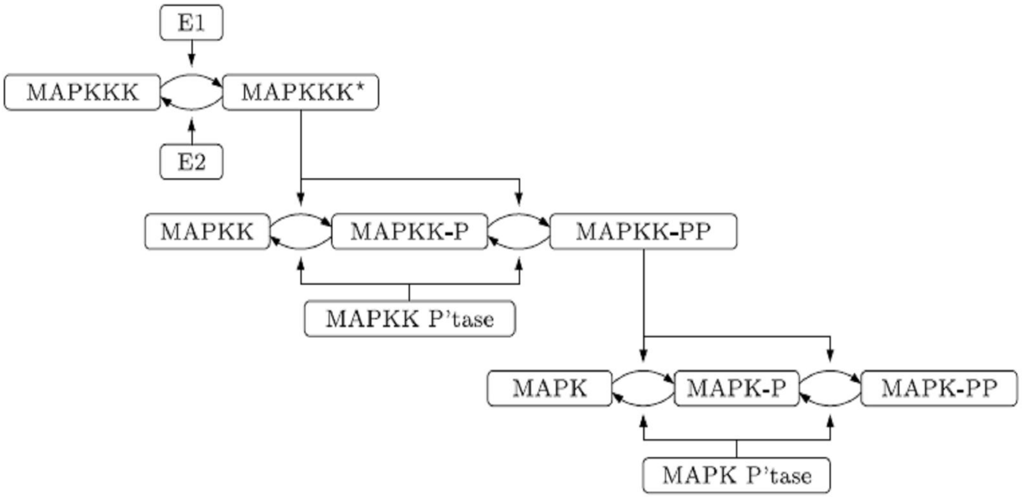


Figure 10: MAPK cascade pathway [16].

In order to construct and analyze a model with PRISM, it must be specified in the PRISM language, a simple state-based language based on the Reactive Modules [17] formalism of Alur and Henzinger. The fundamental components of the PRISM language are modules and variables. A model is composed of a number of modules which can interact with each other. A module contains a number of local variables. The values of these variables at any given time constitute the state of the module. The global state of the whole model is determined by the local state of all modules. The behaviour of each module is described by a set of commands. A command takes the form:

$$[]guard \rightarrow prob\_1 : update\_1 + ... + prob\_n : update\_n;$$

The guard is a predicate over all the variables in the model (including those belonging to other modules). Each update describes a transition which the module can make if the guard is true. A transition is specified by giving the new values of the variables in the module, possibly as a function of other variables. Each update is also assigned a probability (or in some cases a rate) which will be assigned to the corresponding transition [17].

3.6.2 Live Sequence Charts

Live Sequence Charts (LSCs) [18] are an extension of the graphical specification language message sequence charts; notably, they allow a distinction between mandatory and possible behaviour. They have been used successfully by Harel and his co-workers to build visual models of reactive biological systems [19]. By using *play-in* part of the method the application is intuitively learning whereas by using *play-out* the application is executed. It is used in [18] to model C. elegans vulval development where P3.p, P4.p, P5.p, P6.p, P7.p and P8.p are six cells that participate in the formation of vulva. The GUI of the application is shown in Figure 11.

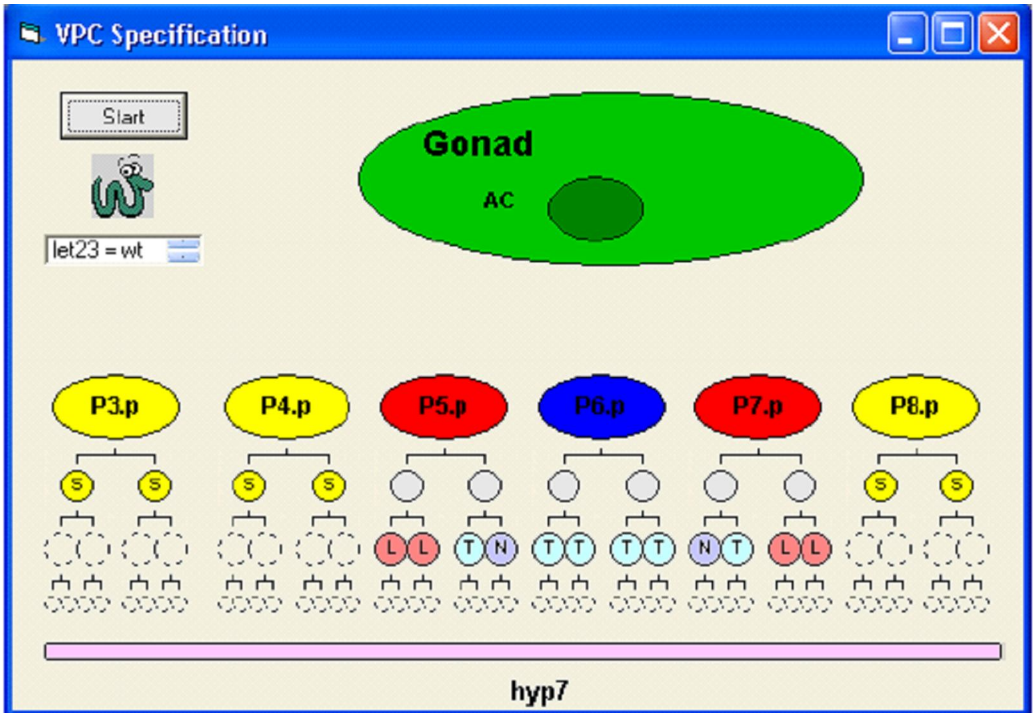


Figure 11: The GUI of Live Sequence Charts [19]

Chapter 4

Petri Nets

4.1 Introduction

Petri Nets (PNs) [2] is a formalism that can help in modelling concurrent and synchronized systems. It is used in Software Engineering but has recently been widely applied in Systems Biology. Molecular biology systems and metabolic networks can be represented in Petri Nets in a natural and convenient way as well as their various extensions – like stochastic, coloured, hybrid and functional Petri Nets.

Petri Nets are a discrete event simulation approach, a case that can help in modelling, analyzing and simulating biological processes. Introduced by Prof. Carl A. Petri in the early sixties, it was mathematical modelling tool to encompass system properties like concurrency, indeterminism, communication and synchronization. Petri Nets can easily be converted into matrices and metrical operations. In Petri Nets the main components are places and transitions. A place can be an entity that has tokens. Tokens are put into the places and represent an attribute of the entity. In the case of molecular biological systems tokens can be the concentration of the molecule, and the molecule is the entity. Places are connected via arcs with transitions and transitions are then connected to the output places. Transitions, under certain conditions (constrains), fire and cause the movement of tokens from the input places to the output places. As shown in Figure 11 places are denoted as circles (p1, p2, p3), transitions as rectangles (t1), tokens are dots that are placed in the circles and arcs are the arrows with a positive integer to show their weight. In the figure they are represented by the variables m1, m2, and m3. All tokens in one place are usually of the same kind. They represent the same attribute. An arc always connects a place with a transition. There is no connection of place to place or transition to transition. Definition 1 giving the formal definition of Petri Nets.

Definition 1: The Petri net *N* is defined by the n-tuple (*P*, *T*, *Pre*, *Post*, *M*)

where:

$P = \{p_1, p_2, ..., p_u\}$, a finite set of places where $u > 0$;

$T = \{t_1, t_2, ..., t_v\}$, a finite set of transitions where $v > 0$, $P \cap T = \varnothing$;

$Pre = P \times T \rightarrow \mathrm{N}$, is the input incidence mapping (weights of the arcs going from places to transitions) and where N is the set of natural numbers;

$Post = P \times T \rightarrow \mathrm{N}$, is the output incidence mapping (weights of the arcs going from transitions to places);

$M = P \rightarrow \mathrm{N}$, is the marking of the net which is a vector of *u* components $(m_1, m_2, ..., m_u)$, where m_i is the number of tokens contained in the place p_i. M_0 is the initial marking [2].

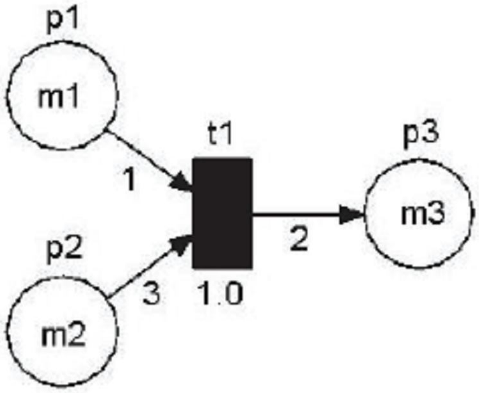


Figure 12: Petri Net [2].

The state of the Petri Net is given by the marking M. The vector M(p) is giving the number of tokens in the place p. If the Pre conditions of the transition are met, the transition will fire. The Pre conditions are usually whether the place has the minimum number of tokens as defined by the weight on the arc. The firing of a transition results a change in marking. Tokens are absorbed from the input place and tokens are created in the output place of the same transition. The number of tokens created in the output place is defined by the Post relation.

In Figure 12 there is an example of how the transition operates. Places $p1$ and $p2$ are input places and place $p3$ is an output place of the transition $t1$. The token contents of places $p1$, $p2$ and $p3$ are $m1$, m2 and $m3$ respectively. The weight constants 1 and 3 on the arcs going out of places $p1$ et $p2$ and the value 1.0 attached to transition $t1$ mean that $t1$ can fire if $m1 \geq 1$ and $m2 \geq 3$, and that the firing delay is 1.0 time unit (in the case of a timed net). When $t1$ is fired, one token is removed from $p1$, three tokens are removed from $p2$ and two tokens are added to $p3$ [2].

Some more definitions of the behaviour and structure of Petri Nets follow.

Definition 2: *Reachability.* A marking M is reachable if it can be reached from the current marking M_i in a finite firing sequence.

Definition 3: *Boundedness.* A place is bounded with bound k, if the token count does not exceed k for any reachable marking M of the net. A Petri net is k-bounded if each place is k-bounded.

Definition 4: *Liveness.* A transition is potentially firable if there exists a sequence of transition firings leading to a marking in which the transition is enabled. A transition is live if it is potentially firable for all reachable markings. A transition is dead if it is not potentially firable at the marking M; so if the Petri net enters marking M, the dead transition cannot fire any more.

Definition 5: *S-invariant.* If C is the incidence matrix corresponding to the result of *Post* — *Pre*, then S-invariants are the solutions to the equation $Cy = 0$. The non-zero entries in the vector y constitute the set of places whose total token count does not change with any firing sequence. It is a conservation rule.

Definition 6: *T-invariant.* If C is the incidence matrix corresponding to the result of *Post* — *Pre*, then T-invariants are the solutions to the equation $CT\ x = 0$, $x \geq 0$. The solution vector *x* is the set of transitions that have to fire, from some marking *M*, to return the Petri net to the same marking *M*. It is a regenerative rule. [2]

In the next sections we present the various extensions of Petri Nets: Functional, Stochastic, Coloured, Hybrid, Hybrid Functional and High-level.

4.2 Functional Petri Nets

Functional Petri Nets introduced a new way to represent the weight of an arc. One can, instead of an integer, put a formula consisting of the marking variables of the places involved in a transition and give a dynamic way the transition performs. In this way, the rate of a chemical reaction can be modified according to the concentrations of the molecules involved.

Definition 7: The functional Petri net *N* is defined by the n-tuple (*P*, *T*, *Pre, Post*, *V*, *M*) where:
(*P*, *T, Pre*, *Post*, *M*) is a Petri net as described in Definition 1;
$V = \{g_a(m_1,...,m_u), a \in \mathrm{Pre} \cup Post \mid g : p_1 \times ... \times p_u \rightarrow \mathrm{N}\}$, a set of functions assigned to arcs of the net using its marking $(m_1, m_2, ..., m_u)$ as parameters [2].

4.3 Stochastic Petri Nets

Stochastic PNs give new dimension to transition timing. Instead of the transitions being instantaneous, they may have a delay that is characterized by a probabilistic distribution. Therefore, the delay is a random variable and the delay mean time is obtained by a stochastic rate .The formal definition is given by Definition 8.

Definition 8: The stochastic Petri net *N* is defined by the n-tuple (*P, T, Pre, Post, F, λ, M*) where:
(*P, T, Pre,Post, M*) is a Petri net as described in Definition 1;

$F = \{F_t, t \in T \mid F_t : [0, \infty) \to [0,1]\}$, a set of probability density functions for the net firing delays. Their average is 1 and they are independent of the marking;
$\lambda = \{\lambda_t, t \in T \mid \lambda_t : \mathrm{N} \to R+\}$, a set of firing rates, which are function of the marking (a set of natural integers) and where each element is associated with a transition *t*. This rate, a positive real number from the set R+, is used to calculate the probability density function for the transition *t* [2].
In Stochastic models the molecule concentrations are discrete amounts and the reaction kinetic rates are random events following probabilistic laws. SPNs are very useful in modelling and simulating such systems and examine the behaviour through the simulation results.

4.4 Coloured Petri Nets

The need to model huge systems led to the use of coloured tokens. This made the models smaller, more manageable and readable. In this way one can examine different dynamic behaviours modelled by different token colours.

Definition 9: The coloured Petri net *N* is defined by the n-tuple (*P, T, Pre, Post, C, M*) where:
(*P, T, Pre, Post, M*) is a Petri net as described in Definition 1 and the tokens of *M* are identified by a colour;
C = *{C*1, *C*2 . . .*}*, a set of colours. The incidence mappings *Pre* and *Post* are functions of the token colours [2].

4.5 Hybrid Petri Nets and Supplementary Extensions

If we consider a token being continuous instead of discrete, then we obtain Hybrid Petri Nets (HPNs). HPNs allow both the continuous and discrete type of tokens. One can have in their model tokens of integer number and tokens of a non-negative real number called marks. In the latter case, transitions must also be continuous with a variable called speed. Speed denotes the rate of quantity transformation from input places to output places.

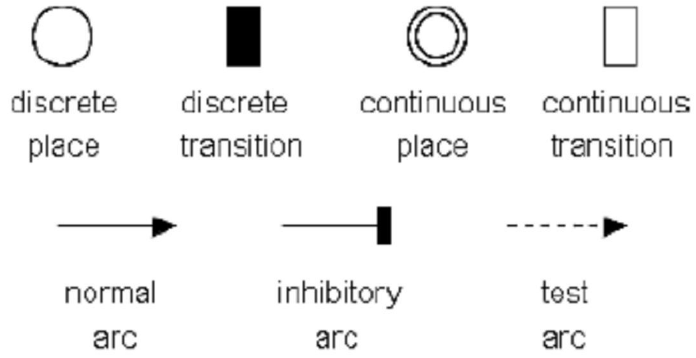


Figure 13: Graphical representations of the elements of Hybrid Petri nets [2].

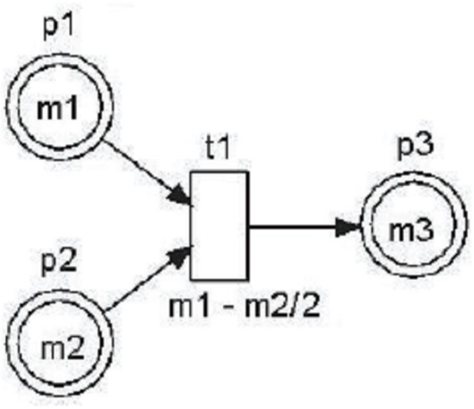


Figure 14: Hybrid Petri Net [2].

Figure 13 shows the graphical representation of elements in Hybrid Petri Nets; introducing the continuous place with double circle, the continuous transition with blank rectangle, inhibitory arc and the test arc. The use of the latter arcs is explained in the next section. In Figure 14 there is an example of how Hybrid Petri Nets work. Places p_1, p_2 and p_3 are continuous places having content m_1, m_2 and m_3 respectively. The function $m_1 - m_2/2$ is assigned to the continuous transition t_1 as its firing speed, t_1 can be fired if

$m_1 > 0$ and $m_2 > 0$. The contents of p_1 and p_2 are consumed with the speed $m_1 - m_2/2$, and the content of p_3 increases with the same speed when the transition t_1 is fired [2].

Definition 10: The hybrid Petri Net N is defined by the n-tuple *(P, T, Pre, Post, h, M)* where:

(P, T, Pre, Post, M) is a Petri Net as described in Definition 1, where M is a combination of integers for the number of tokens in discrete places and of real numbers for the mark of continuous places;

$h: P \cup T \to \{D, C\}$, called a hybrid function, indicates for each place and transition, if it is discrete $(h(p_i) = D$ and $h(t_j) = D)$ or continuous $h(p_k) = C$ and $h(t_1) = C)$;

A delay d_t is assigned to all discrete transition and a speed u_t is assigned to all continuous transitions [2].

4.6 Hybrid functional Petri net (HFPN)

Combining the functions of Hybrid and Functional Petri Nets, a new formalism was born called Hybrid Functional Petri Nets (HFPN). With HFPNs not only one can use a formula on the arcs to notify the weight, but two more kinds of arcs were added. The inhibitory arc, first, that denotes the repression of one place to another. When using an inhibitory arc the place must have less or equal number of tokens to the weight on the arc, in order to enable the transition to fire. The second kind of arc is the test one. This arc is used in the case that the transition does not consume any tokens. It is appropriate for modelling chemical reactions where an enzyme is needed but not consumed by the reaction.

4.7 High-Level Petri Nets

High level PN consist of places, transitions and arcs. The difference from the other PN is that tokens take actual values from a data set. In the case of our Figure 15 the tokens in p1 can take values from the set {0..5}. Similarly p2{0..5} and p3{0..10}. Like in sets, places can have multiple tokens with the same value. In p1 we can have two 3's or three 4's. The marking of all places denotes the state of the Petri Net. In this case marking M is M(p1) = {1}, M(p2) ={2,3} and M(p3) = {}.

The second difference from the other PN is the variables on the arc (a, b, c) that give the binding of tokens to the transitions. Transitions also have a Boolean expression called guard. The transition fires if guard evaluates to true or if the binding enables the transition, which means that the arc variable has the same value as the token in the place. In our example, the transition is enabled by the marking: $a \mapsto 1$, $b \mapsto 3$, $c \mapsto 2$.Tokens 1 and 3 are moved from places p1 and p2 respectively and a new token (2) is created in p3. the new marking is M'(p1) = {}, M'(p2) = {2}, M'(p3) = {2}. A reachability graph can be formed based on the initial marking, giving all the reachable states a HLPN can reach. Model checking has several techniques in analyzing reachability properties.

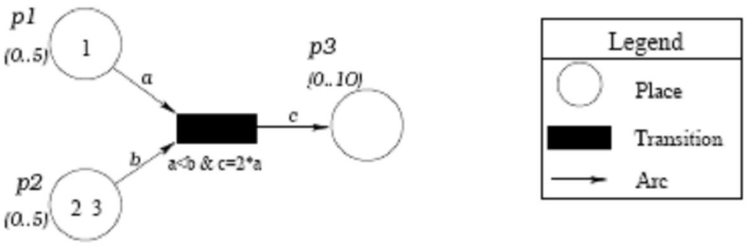


Figure 15: An example of a simple high-level Petri net [20].

Definition 10: A high-level Petri net, HLPN for short, is a triple (S ,T ,i), where S and T are disjoint sets of places and transitions, and ι is an inscription function with domain $S \cup (S \times T) \cup (T \times S) \cup T$ such that:

- – for every place $s \in S$, $i(s) \subseteq Val$, is the type of s , i.e., the set of possible values the place may carry;
- – for every transition $t \in T$, $i(t)$ is the guard of$t$, i.e., a predicate from Pr;
- – for every arc $(s,t) \in (S \times T) : \iota((s,t)) \in M_f(Val \cup Var)$ is a multi-set of variables or values analogously for arcs $(t,s) \in (T \times S)$. The inscriptions

$i((s,t))$ and $i((t,s))$ will generally be abbreviated as $i(s,t)$ and $i(t,s)$, respectively. The arcs with empty inscriptions are omitted. [21]

4.8 Evaluation

4.8.1 Qualitative vs. Quantitative Analysis

Depending on the aspect one takes on biological systems, Petri Nets can be used for qualitative or quantitative analysis. If one is more concerned for the dynamics of the system, the biochemical reactions that take place, the boundaries or liveness of the system then the Petri Net extension should help for the qualitative analysis of the system. On the other hand, if one wants to know the concentrations of the molecules involved or the reachable steady states of the system then the Petri Net extensions should help for the quantitative analysis of the biological system (Table 2).

4.8.2 Hybrid vs. Stochastic Petri Nets

The choice of whether to use a Hybrid or Stochastic Petri Net relies on the nature of the system. If the biological system consists of small amount of molecules, the better choice is stochastic - along with the case where your transitions act upon timed-distributions. If, however, the system is composed of large amount of molecules then the answer is hybrid Petri Net - taking also into consideration that your places and transitions are continuous.

Table 2: Summary of the capabilities and goals of each type of Petri Net

Petri Net Extension	**Modelling Goal**	**Pros**	**Analysis Type**	**Available Software**	**Good for**
Coloured	Analysis of biological system properties	Able to diminish model size, allow models to manage more information without being complex. Can represent in the same model, different dynamic behaviours modelled by different token colours.	Qualitative	Design/CPN	Discriminating metabolites on the basis of their chain of reactions in a model
Stochastic	Simulation of biological systems with low concentrations	Can take into consideration the delay of the reaction, which follows a probabilistic distribution	Quantitative	Mobius	Model with small number of molecules
Functional	Simulations of biological reactions	Can assign to an arc an equation using marking variables. Concentrations, represented by the number of tokens in			The network marking dynamically modifies the weight of the arc

		the net, are variables for the functions that define the weight of the arc			
Hybrid	Simulation of biological systems	Can represent discrete and continuous quantities	Quantitative	Genomic Object Net (renamed to Cell Illustrator)	Model with high number of molecules
High-Level	Model multi-valued networks	Tokens can have a value within a range of values, resulting a multi-value representation			Model the dynamics of a transcription networks with the different concentrations of the reactants

4.9 Available tools

There are several tools available that model biological systems. Some of them are shown in Table 2. Cell Illustrator, Mobius and PEP tool are examples. In the next sections we give an overview of Mobius and a brief description of PEP tool.

4.9.1 Mobius

Mobius is the tool that we used in our case study. It is software developed by the University of Illinois and we express our sincere thanks to them for allowing us to use it. It is a Java-based, cygwin application that provides the ability to build your model in SAN (Stochastic Activity Network); a Stochastic Petri Net formalism. The tree structure of a project in Mobius is shown in Figure 16.

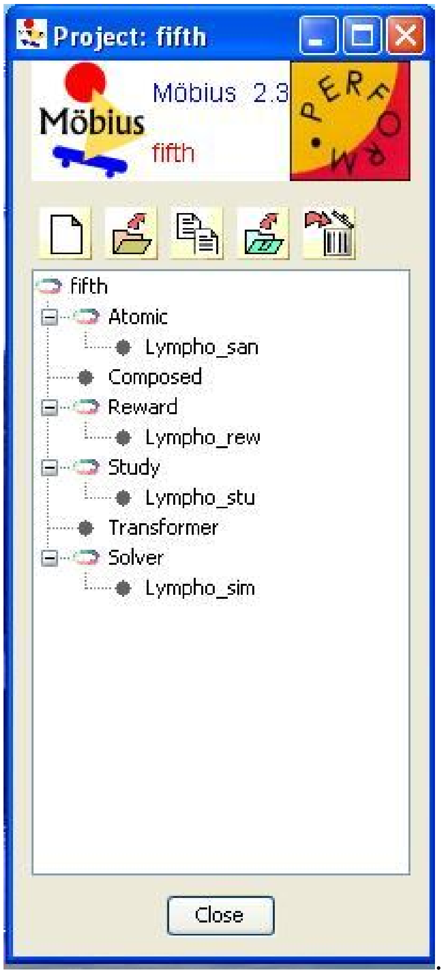


Figure 16: The tree structure of Mobius Project

The first component is the atomic formalism where the .san file (Figure 17) is built based of the following primitives (elements):

- place, represented by a blue circle.
- extended place, represented by an orange circle.
- input gate, represented by a red triangle with its tip pointing to the left.
- output gate, represented by a black triangle with its tip pointing to the right.
- instantaneous activity, represented by a thin vertical bar in blue colour
- timed activity, represented by a thick vertical bar in blue colour

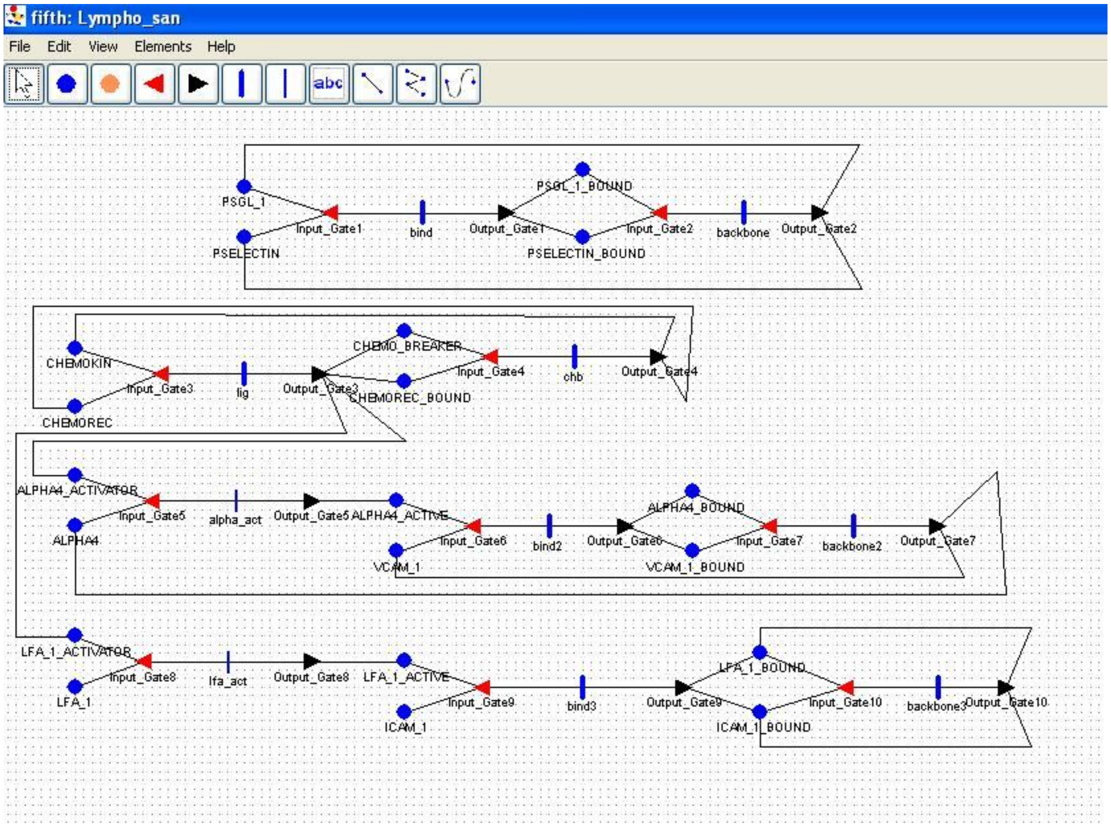
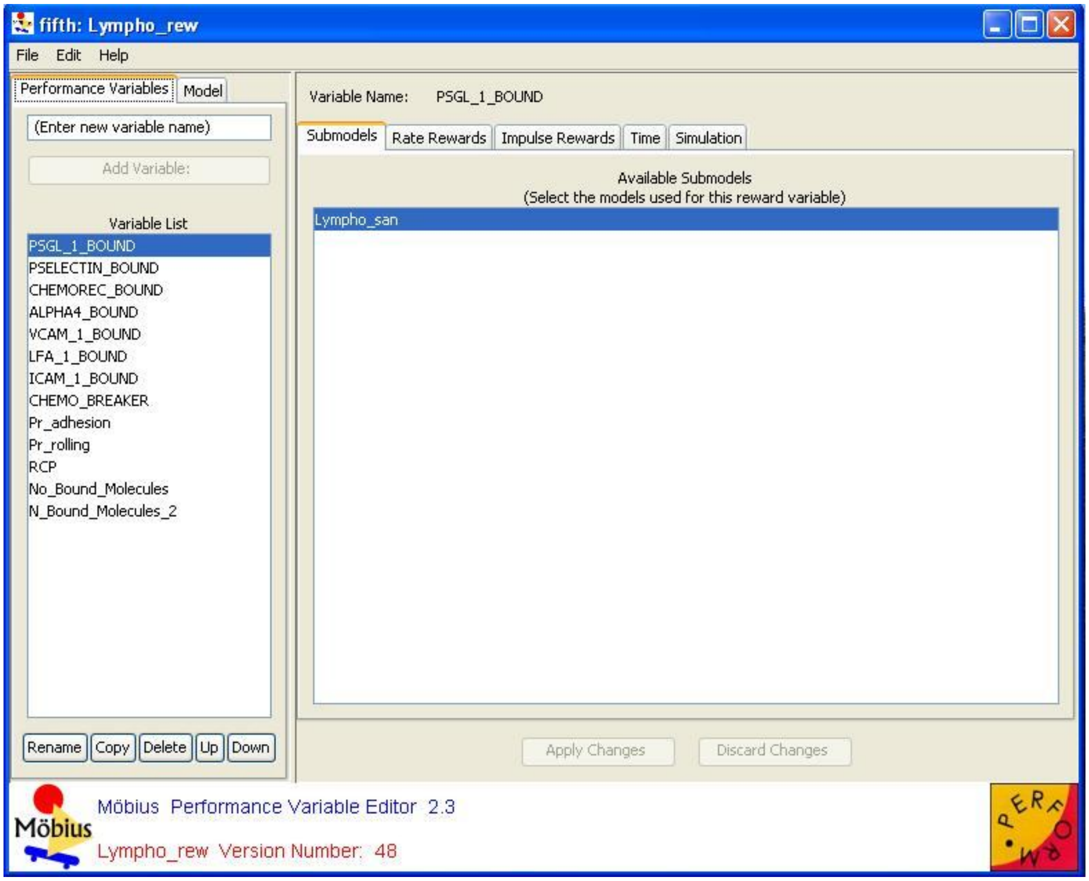


Figure 17: The SAN formalism of Mobius

Atomic formalisms can be joined to make composed formalisms with the Join/Replicate procedure.

The next component in the project is the reward model (Figure 18). Here one can declare the reward performance variables (PV). These are variables that we are interested in their values during the simulation. One can add a PV in the left hand side column and set the reward function in the rate reward tab, by which function one can specify what should the performance variable return. For example in the case of the variable PSGL_1_BOUND we are interested in the concentration of PSGL-1_BOUND and the reward function returns the marking of this place. More over, one can specify in the Time tab the Type of time, whether the variable is tracked at an instant of time or at an interval of time or at time average interval or at steady state. Below this one defines the upper bound of time, the starting point of time and the pace by which the PV is tracked.



(Enter new variable name)

Variable Name: PSGL_1_BOUND

Variable List

PSGL_1_BOUND
PSELECTIN_BOUND
CHEMOREC_BOUND
ALPHA4_BOUND
VCAM_1_BOUND
LFA_1_BOUND
ICAM_1_BOUND
CHEMO_BREAKER
Pr_adhesion
Pr_rolling
RCP
No_Bound_Molecules
N_Bound_Molecules_2

Available Submodels
(Select the models used for this reward variable)

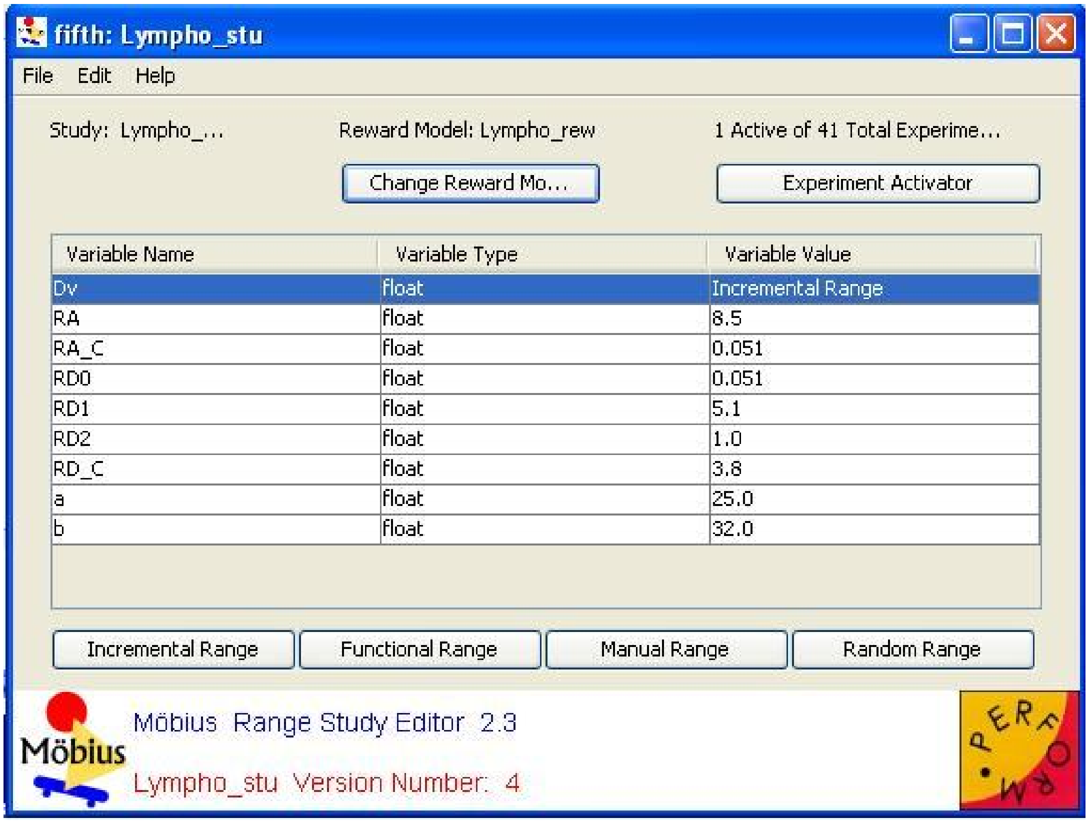
Lympho_san

Möbius Performance Variable Editor 2.3
Lympho_rew Version Number: 48

Figure 18: The reward model in Mobius

In the Simulation tab one can check the reward estimation; meaning whether one likes Mobius to estimate the mean of the PV, the variance, the interval or the distribution. On the right-hand side one can specify the confidence intervals of the PV. This means that if the PV exceeds the confidence intervals then the user should be notified by the results. In the Excel files Mobius provides, an asterisk is shown if such thing happens.

The third component of the tree is the study (Figure 19). Here one can specify the constants and variables that the simulation uses. Mobius provides the facility of defining a variable with different values ranging in incremental, functional, manual or random way. In our example Dv (vessel diameter) is defined as such variable that changes in incremental range. In the study one declares the type of the variable (float, integer) and defines the value of the variable. If one variable is taking different values, then Mobius creates one experiment for each value of this variable. In our case here 41 experiments were created for each value of the vessel diameter.



Study: Lympho_...

Reward Model: Lympho_rew

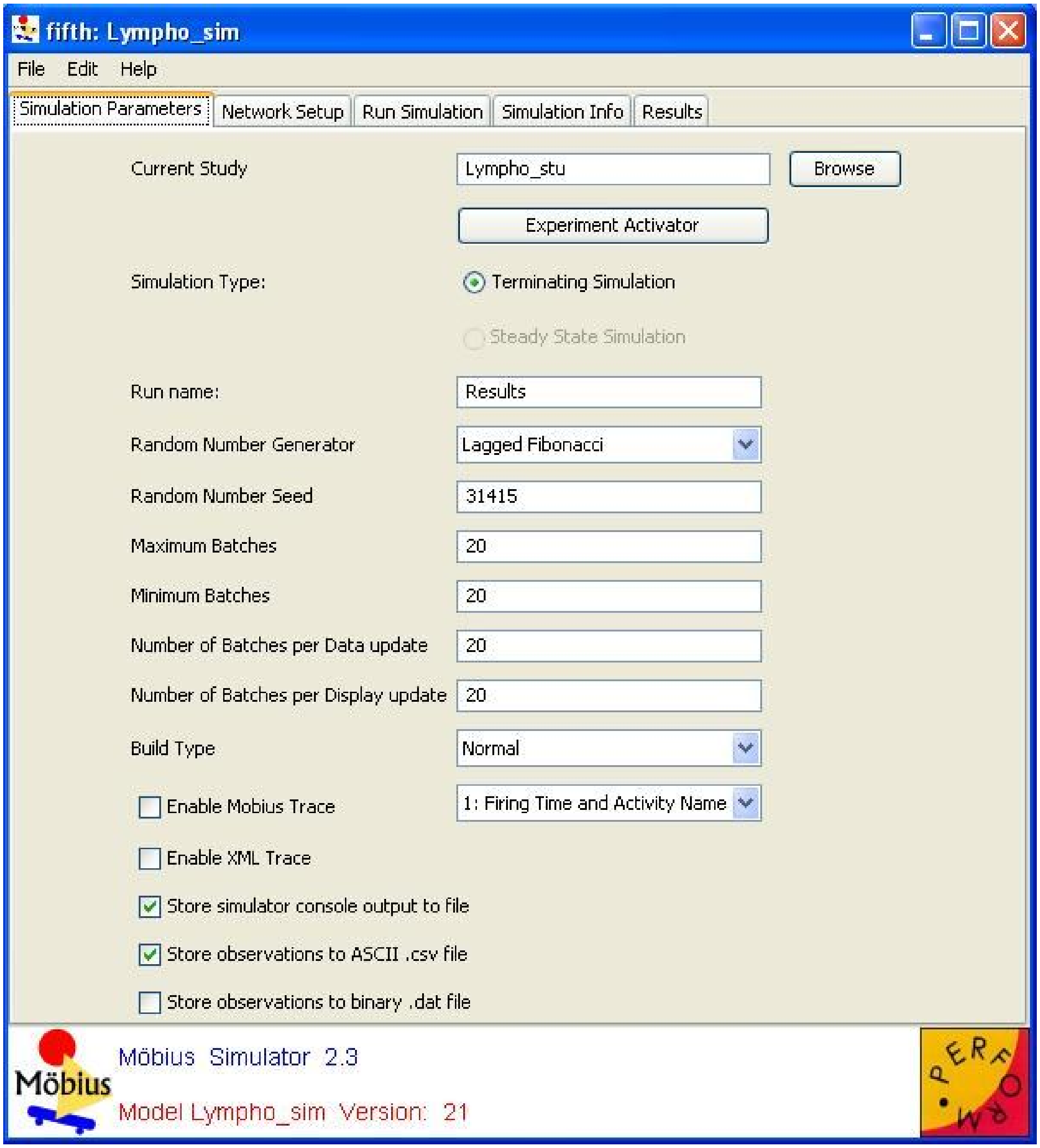
1 Active of 41 Total Experime...

Variable Name	Variable Type	Variable Value
Dv	float	Incremental Range
RA	float	8.5
RA_C	float	0.051
RD0	float	0.051
RD1	float	5.1
RD2	float	1.0
RD_C	float	3.8
a	float	25.0
b	float	32.0

Lympho_stu Version Number: 4

Figure 19: The study in Mobius

The fourth and final component in the project tree is the solver (Figure 20) in the first tab of which one defines the simulation parameters. This means the current study that will be executed, the experiments that will be executed, the simulation type (terminating simulation or steady state), the name of the result files, the random number generator (Lagged Fibonacci or Tausworthe), the random number seed, the maximum and minimum batches, the number of batches per data update, the number of batches per display update, the built type, and the format of result files that will be created.



Current Study: Lympho_stu

Simulation Type: ● Terminating Simulation ○ Steady State Simulation

Run name: Results

Random Number Generator: Lagged Fibonacci

Random Number Seed: 31415

Maximum Batches: 20

Minimum Batches: 20

Number of Batches per Data update: 20

Number of Batches per Display update: 20

Build Type: Normal

□ Enable Mobius Trace — 1: Firing Time and Activity Name

□ Enable XML Trace

☑ Store simulator console output to file

☑ Store observations to ASCII .csv file

□ Store observations to binary .dat file

Figure 20: The solver in Mobius

One can place Mobius on a network where various developers can work on the same project. This can be done in the Network setup tab. The Run Simulation tab is where the simulation is executed. By checking on the “Start Simulation” button Mobius starts compiling the project hierarchically first the SAN model, then the reward model, then the study and finally the solver. The whole model is converted by Mobius to C++ code, compiled and then runs the simulation. By the end of the simulation, the results can be found in the Results tab, or in the MobiusProject folder in C drive (e.g. C:\MobiusProject\fifth\Solver\Lympho_sim). In our example the result files will be in Excel format.

In Figure 21 it shown how the various parts of Mobius project are connected, compiled, linked and run to give the results.

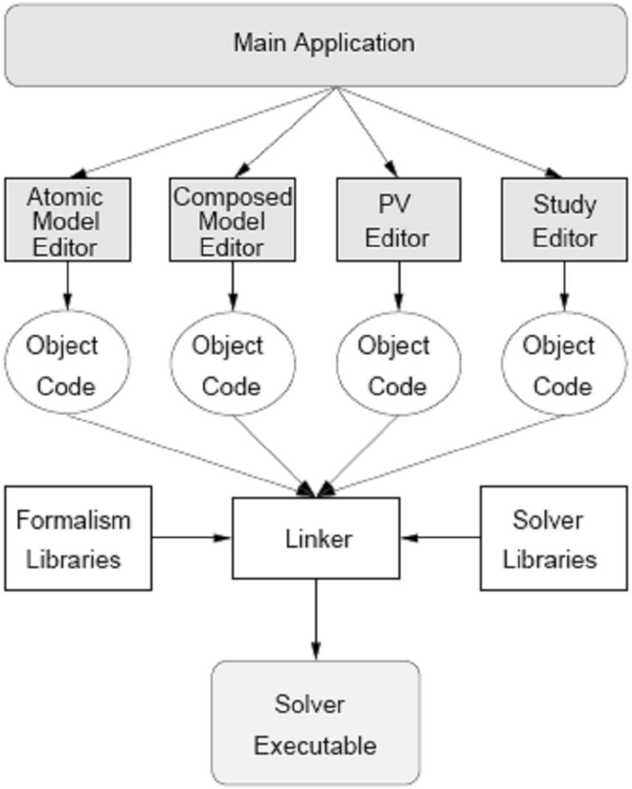


Figure 21: Mobius tool architecture [22].

4.9.2 PEP tool

The PEP tool (Programming Environment based on Petri Nets) is a comprehensive set of modelling, compilation, simulation and verification components, linked together within a Tcl/Tk-based graphical user interface. PEP's modelling components facilitate the design of parallel systems by parallel programs (B(PN)^2 and SDL), interacting finite automata, process algebra, or high-level/low-level Petri nets. PEP's compilers generate Petri nets from such models. Its simulators allow automatic or user-driven simulation of high-level / low-level nets and may trigger simulation of the corresponding programs and/or a 3D model. PEP's verification component contains various Petri net indigenous algorithms to check, e.g., reachability properties and deadlock-freeness, as well as verification algorithms.

Chapter 5

Case Study

5.1 Description

The system tries to simulate the recruitment of lymphocytes during an inflammation in brain vessels. The procedure can be considered as a cause of multiple sclerosis in neural system. Multiple sclerosis (abbreviated MS, also known as disseminated sclerosis or encephalomyelitis disseminata) is a disease in which the fatty myelin sheaths around the axons of the brain and spinal cord are damaged, leading to demyelization and scarring as well as a broad spectrum of signs and symptoms. MS affects the ability of nerve cells in the brain and spinal cord to communicate with each other. Nerve cells communicate by sending electrical signals called action potentials down long fibers called axons, which are wrapped in an insulating substance called myelin. Almost any neurological symptom can appear with the disease, and often progresses to physical and cognitive disability. MS takes several forms, with new symptoms occurring either in discrete attacks (relapsing forms) or slowly accumulating over time (progressive forms). MS is an autoimmune disease that its pathophysiology is still under investigation [23].

Our work is a simulation of the recruitment of lymphocytes during an inflammation of brain vessels. The system examines the four phases of the recruitment: tethering of the cell on the endothelium of the vessel, rolling of the cell on the endothelium, chemokine activation and firm adhesion of the cell by the vessel (Fig. 22). The latter causing the diapedesis of the cell to the parenchyma of the nervous system. There are certain molecules that play vital role to the adhesion of the cell. On the lymphocyte there are: PSGL-1, ALPHA4, CHEMOREC and LFA-1 integrins that form bonds with the corresponding molecules on the endothelium, ligands: P-SELECTIN, VCAM-1, CHEMOKINE and ICAM-1. The bond is a heterodimer complex that is like key to lock bondage. This bondage turns each molecule into a BOUND form, i.e. PSGL_1_BOUND, ALPHA4_BOUND, P_SELECTIN_BOUND. The first phase is the interaction of PSGL-1 and P-SELECTIN, helping in the tethering of the cell with the endothelium. The second phase is the integrin activation by the interaction of CHEMOKINE and CHEMOREC. G-protein is produced during this phase and activates integrins and ligands of the next phases. The bondage of ALPHA4 and VCAM-1 is the third phase, contributing to the adhesion of the cell to the endothelium. The final phase is the interaction of LFA-1 and ICAM-1, which makes the firm adhesion of the cell and the final diapedesis. Diapedesis means that the cell crosses the Blood Brain Barrier (BBB) and intrudes parenchyma and nervous system. In such case, lymphocytes can make cause demyelization of the neural axons leading to miscommunication of the neural cells and muscular dysfunction. Anatomically this is shown in MRI's as white lesion.

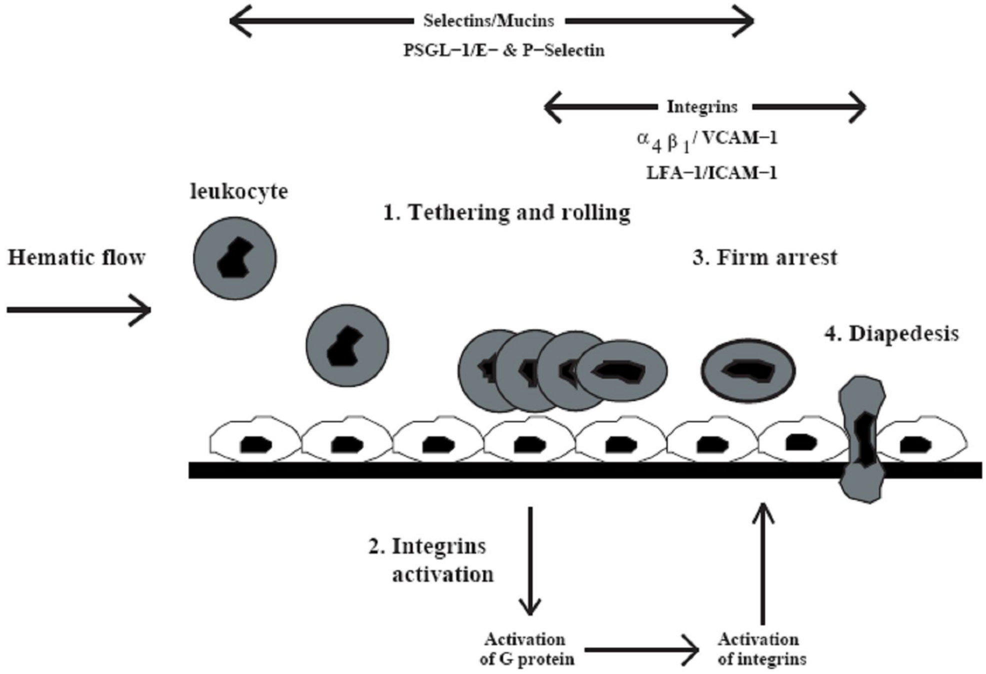


Figure 22: The four-phase model of lymphocyte recruitment [24].

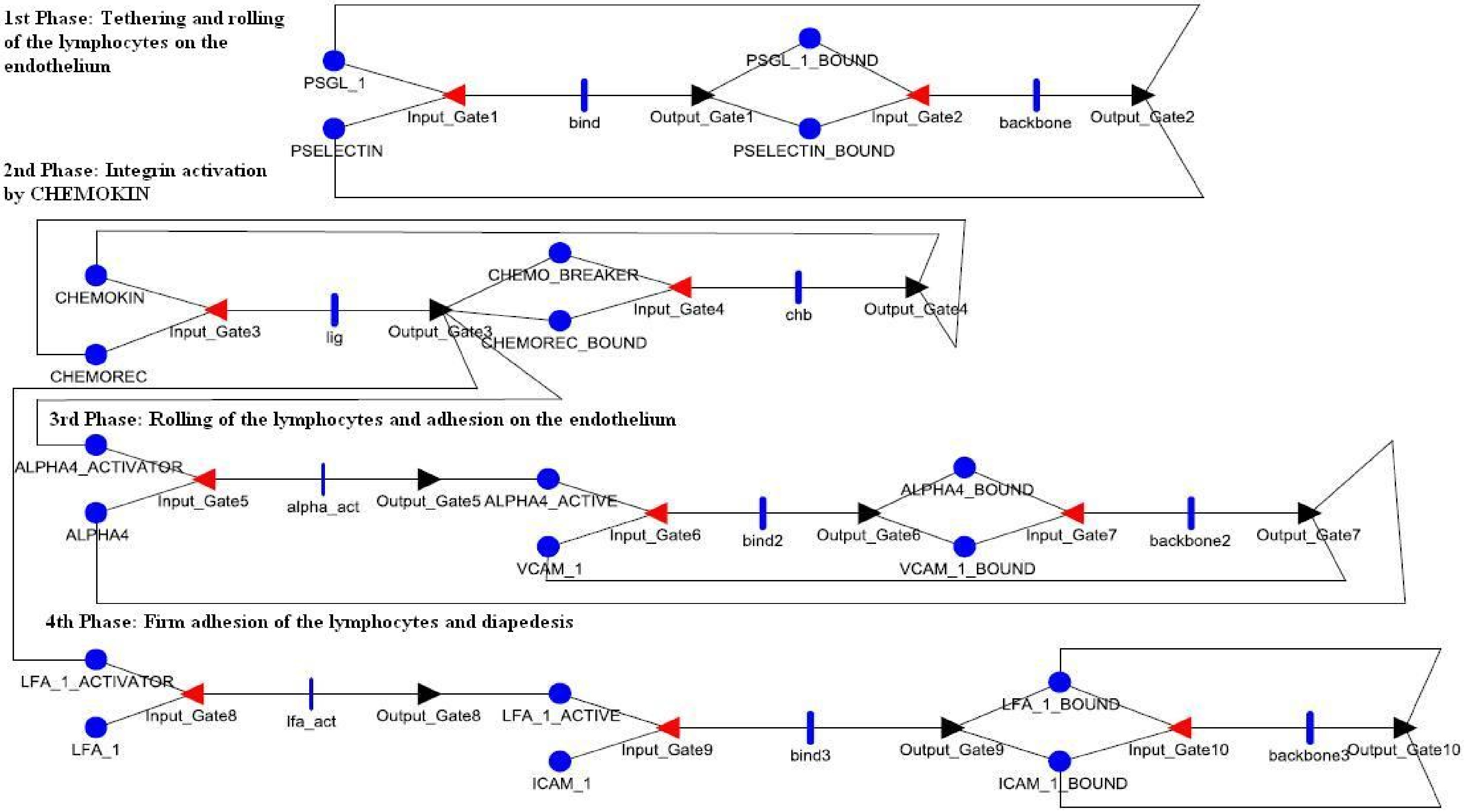


Figure 23: The SAN formalism of the system using Mobius tool.

For our simulation we used the Mobius tool version 2.3 from the University of Illinois and we gratefully thank them for providing us the software. The system was modelled in SAN (Stochastic Activity Network) formalism. This is actually Stochastic Petri Net formalism that can be drawn in Mobius and the tool turns it into C++ code simulation.

The setup data for our simulation are shown in Tables 3, 4.

Table 3: Space parameters and densities [24]

Space Parameters for Vessels and Lymphocytes			
Radius of vessel		25μm	
Length of vessel		100μm	
Volume of vessel		$1{,}96\text{x}10^5\mu m^3$	
Radius of lymphocyte		5μm	
Densities			
Lymphocyte Molecules	**Endothelium Molecules**	**Density in $\mu m\text{-}^2$**	**Number of Molecules ($\text{x}10^6$)**
PSGL-1	P-SELECTIN	$5600\mu m\text{-}^2$	88
ALPHA4	VCAM-1	$85\mu m\text{-}^2$	2
CHEMOREC	CHEMOKINE	$15{,}000\mu m\text{-}^2$	236
LFA-1	ICAM-1	$5500\mu m\text{-}^2$	86

The simulation data are based on mice experiments. The volume of the simulation and the radius of the vessels and the radius of the cells were taken from experimental data on mice. The mice were affected by experimental autoimmune encephalomyelitis which is the analogous disease to multiple sclerosis in human beings. The association of ligands and integrins is characterized by a rate of association and the corresponding disassociation by a rate of disassociation. The values of these rates are shown in Table 4.

Table 4: The values of Rates [24]

RA = 8.500	RA_C = 0.051	RD_0 = 0.051
RD_1 = 5.100	RD_2 = 1.000	RD_C = 3.800

RA = Rate of association of PSGL-1 and P-SELECTIN, rate of association of ALPHA4 and VCAM-1, rate of association of LFA-1 and ICAM-1.

RA_C = Rate of association of CHEMOKINE and CHEMOREC

RD_0 = Rate of disassociation of PSGL-1 and P-SELECTIN

RD_C = Rate of disassociation of CHEMOKINE and CHEMOREC

RD_1 = Rate of disassociation of ALPHA4 and VCAM-1

RD_2 = Rate of disassociation of LFA-1 and ICAM-1

The first phase of the recruitment of lymphocytes is the tethering of lymphocytes to the endothelium. To this purpose the molecules that contribute are PSGL-1 and P-SELECTIN. As shown in Figure 23, PSGL_1 and PSELECTIN pass through the *bind* transition and are turned into PSGL_1_BOUND and PSELECTIN_BOUND respectively. Bind transition is a timed-activity denoted by a thick vertical line and is characterized by exponential distribution. The rate of the distribution is

*RA * concentration of PSGL-1 * concentration of P-SELECTIN*

The concentration of PSGL-1 is $88x10^6$ molecules and the concentration of P-SELECTIN is also $88x10^6$ (Table 3). During the *bind* transition the input places are decreased in tokens by one while the output places (PSGL-1_BOUND and PSELECTIN_BOUND) are increased by one. Input Gates generally are responsible for the changes that take place in the marking of input places when the transition fires. Input gates take two input fields; the predicate and the input function. Predicates are boolean equations that have to be true in order the transition to fire. Input function determines how the number of tokens in the input places is changed after the transition fires. In the case of *bind* transition the predicate in the input gate is that the marking of the input places should be greater than zero. The output gate is responsible for the flow of tokens after the transition is fired. It adds one token to each output place. Output gates take one field, the output function.

Not all the cells however stay tethered on the endothelium. The heterodimer that is formed can break and give the molecules back their free state. PSGL_1_BOUND and PSELECTIN_BOUND go through the transition *backbone* that is also a timed-activity characterized by exponential distribution. Input gate checks that the markings of the input places are greater than zero and decreases the tokens by one. The rate of distribution is:

*RD0 * concentration of PSGL-1_BOUND * concentration of P-SELECTIN_BOUND*

The Output_Gate_2 increases the marking of PSGL-1 and P-SELECTIN by one. PSGL-1 and P-SELECTIN play role in tethering but also in rolling of the cell. So as the cell rolls on the endothelium the leading part of the cell creates bonds while the trailing part of the cell breaks bonds.

In the second phase we have the CHEMOKIN interacting with the CHEMOREC. CHEMOKIN is a chemotactic cytokine in the endothelium cells, while CHEMOREC are receptors that reside on the lymphocyte. The bondage between the two promotes the production of G-protein and the activation of integrins for the following phases. CHEMOKIN and CHEMOREC interact via *lig* transition and the products are: CHEMOREC_BOUND and CHEMOKIN_BOUND. CHEMOKIN_BOUND turned into: CHEMO_BREAKER, ALPHA4_ACTIVATOR and LFA_1_ACTIVATOR. *lig* transition is a timed activity with exponential distribution. All timed activities in the model are characterized by exponential distribution. The rate of the distribution in the chemokin activation is:

*RA_C * concentration of the CHEMOKIN * concentration of CHEMOREC*

Input_Gate_3 predicate makes sure that the concentrations of CHEMOKIN and CHEMOREC are greater than zero, while the input function reduces the tokens in CHEMOKIN place and CHEMOREC place by one. The Output_Gate_3 increases the marking of CHEMOREC_BOUND, CHEMO_BREAKER, ALPHA4_ACTIVATOR and LFA-1_ACTIVATOR by one.

CHEMOREC_BOUND and CHEMO_BREAKER go through the *chb* transition. *chb* transition is a timed activity that checks that the concentration of CHEMOREC_BOUND and CHEMO_BREAKER are greater than zero through Input_Gate_4. Input_Gate_4 reduces the tokens in the input places by one, while Output_Gate_4 increases the tokens of the output places, CHEMOKIN and CHEMOREC, by one. The rate of the probabilistic distribution in the *chb* transition is:

*RD_C * concentration of CHEMO_BREAKER * concentration of CHEMOREC_BOUND*

The ALPHA4_ACTIVATOR molecule that was produced in the previous phase, contributes in the third phase of the recruitment. ALPHA4_ACTIVATOR activates ALPHA4 through the *alpha_act* transition. Alpha_act transition is an instantaneous transition. This means that it has no time delay and no stochasticity. The transition fires instantaneously as soon as the predicate in Input_Gate_5 is satisfied. Instantaneous transitions are denoted by a thin vertical line. The rate is infinite (A). Input_Gate_5 predicate checks whether the concentrations of ALPHA4_ACTIVATOR and ALPHA4 are greater than zero and decreases the marking of the input places by one. Output_Gate_5 increases the marking of the output place ALPHA4_ACTIVE by one. This phase is responsible for the rolling of the cell and the adhesion of the cell by the endothelium. ALPHA4_ACTIVE interacts with the VCAM-1 molecule that resides on the endothelium to form a heterodimer complex. The transition *bind2* makes sure that the concentration of the two input places are greater than zero and once enabled reduces the marking of each place by one. bind2 transition is a timed activity and the rate of the probabilistic distribution is:

*RA * concentration of ALPHA4_ACTIVE * concentration of VCAM-1*

Output_Gate_6 increases the marking of the output places, ALPHA_BOUND and VCAM-1_BOUND, by one token.

ALPHA_BOUND and VCAM-1_BOUND return to their free states via the *backbone2* transition. The breakage of their bond is possible as the cell rolls on the endothelium especially if the adhesion did not happened. backbone2 is a timed activity with probabilistic distribution. The rate of disassociation is RD_1 and the rate of the exponential probabilistic distribution is:

RD_1 ** concentration of ALPHA_BOUND * concentration of VCAM-1_BOUND*

The fourth and final phase of the recruitment is the firm adhesion of the cell on the endothelium and the diapedesis into the parenchyma. The molecules that contribute to this are LFA-1 and ICAM-1. LFA-1_ACTIVATOR is responsible for the activation of LFA-1. LFA_ACTIVATOR and LFA-1 interact via the lfa_act transition. It is an instantaneous transition with an infinite rate (A). Input_Gate_8 predicate checks that the concentrations of the two input places are greater than zero and the input function reduces the marking of LFA_ACTIVATOR and LFA-1 by one. Output_Gate_8 increases the marking of LFA-1_ACTIVE by one token.

LFA-1_ACTIVE interacts with the ICAM-1 molecule to form a heterodimer. This is done via the *bind3* transition. The input predicate function of Input_Gate_9 checks that the input places have concentrations greater than zero and the input function reduces the tokens of the input places by one. The rate of the exponential distribution of the timed-activity bind3 is:

*RA * concentration of LFA-1_ACTIVE * concentration of ICAM-1*

Output_Gate_9 increases the marking of the output places, LFA-1_BOUND and ICAM-1_BOUND, by one in the output function. Once the lymphocyte is firmly adhered, it stays adhered until the final diapedesis in the parenchyma. This is shown in Figure 22 with the transition *backbone3* that does not return the bound molecules to their free states, but stay in the states LFA-1_BOUND and ICAM-1_BOUND. Input_Gate_10 decreases the marking of the input places LFA-1_BOUND and ICAM-1_BOUND, but the Output_Gate_10 returns the tokens to the bound states and not to the free states. The rate of backbone3 is given by the equation:

*RD2 * concentration of LFA-1_BOUND * concentration of ICAM-1_BOUND*

The system of PSGL-1/PSELECTIN interaction works in parallel with the rest of the system, a fact that is a privilege of Stochastic Petri Nets and Mobius. This form of parallel procedures is really helpful in Systems Biology, which in most of the times are not sequential and are non-deterministic. In real life many things are done at the same time, out order and with a possibility of being done. Stochastic Petri Nets gives the facility to model biological systems with more real life aspect.

5.2 Results - Analysis

Using the Mobius tool, version 2.3 we set up out simulation based on the data shown in Table 3. The simulation ran for 10 seconds in a vessel of radius 25μm, length 100μm, and a volume of $1{,}96\text{x}10^{5}\mu\text{m}^{3}$. The radius of the lymphocyte is 5μm and the concentrations of the molecules are $88\text{x}10^{6}$ for PSGL-1 and P-SELECTIN, $236\text{x}10^{6}$ for CHEMOKINE and CHEMOREC, $2\text{x}10^{6}$ for ALPHA4 and VCAM-1, and $86\text{x}10^{6}$ for LFA-1 and ICAM-1. We tracked the performance reward variables we declared. Performance reward variables in Mobius are variables that we are interested in their values during the execution of the simulation. In our case we declared the following reward variables:

- PSGL-1_BOUND
- PSELECTIN_BOUND
- CHEMO_BREAKER
- CHEMOREC_BOUND
- ALPHA4_BOUND
- VCAM-1_BOUND
- LFA-1_BOUND
- ICAM-1_BOUND
- Pr_adhesion
- No_bound_molecules

We are interested in the concentrations of the bound molecules of each species of molecules, in the probability of adhesion (Pr_adhesion) and finally in the total number of bound molecules (No_bound_molecules) which is all the bound molecules added up.

The probability of adhesion [24] is calculated with the formula:

$$\Pr(adhesion) = \frac{1}{N_l}\sum_{i} w_i N_i \qquad (11)$$

where $N_l = S_{endothelium} / S_{contact}$ is the total number of lymphocytes on the laminar flux in contact with the endothelium, given by the ratio between the endothelial surface (~15,700 μm^2) and the cell contact area (~200μm^2); N_i is the number of bound molecules for the *i*th molecular interaction; and *w*s are the weights of the linear model that quantify the statistical influence of the different molecular interactions in the cell adhesion mechanism. In our model, the weights can take values in the range between 0 and 1. Because of the lack of experimental quantifications for the statistical influence of the different molecular interactions, we assume that $w_i = 1/8 = 0.125$ for all the considered interactions of all the eight molecular species.

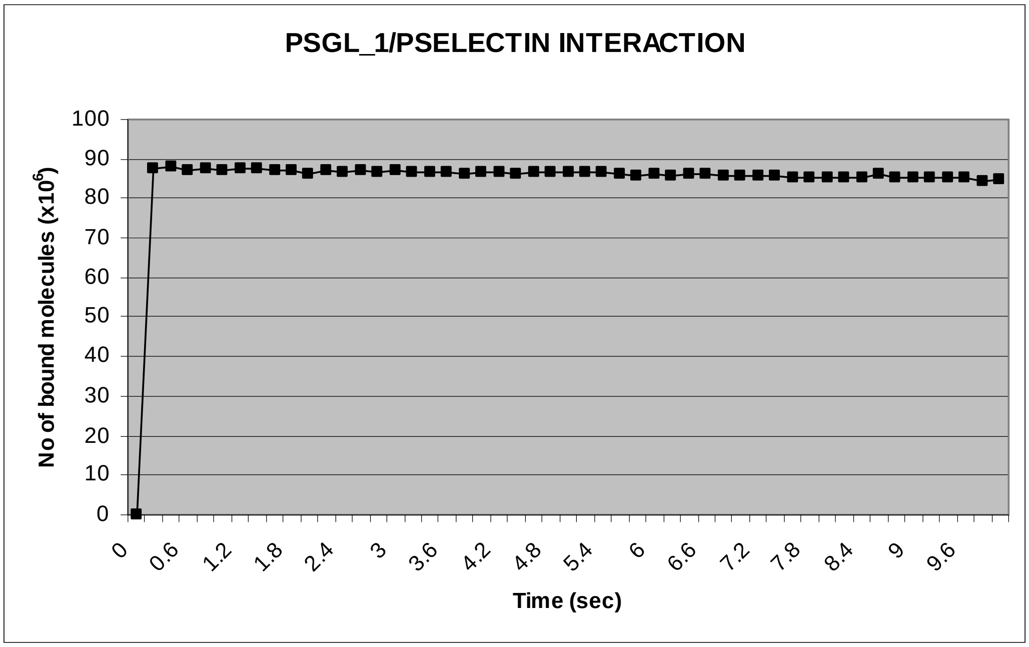


Figure 24: PSGL-1/P-SELECTIN interaction. Number of bound molecules (x10^6) PSLG-1_BOUND/P-SELECTIN_BOUND during the first phase of the lymphocyte recruitment.

In Figure 24 the y-axis represents the number of PSGL-1_BOUND/P-SELECTIN_BOUND molecules in millions during the first phase of the lymphocyte recruitment. The x-axis is the simulation time for which the experiment is executed (10 sec). The results show that the PSGL-1/PSELECTIN interaction (Figure 24) has a steep rise at the beginning of the simulation reaching the 87.9x10^6 bound molecules. This is because there is a high rate of association and a low rate of disassociation. The steep rise is followed by a small fluctuation of the bound molecules that is caused by the forming and breaking of the bonds between the cell and the endothelium. The values of PSGL-1_BOUND and P-SELECTIN_BOUND range from 84.3x10^6 to 87.9x10^6.

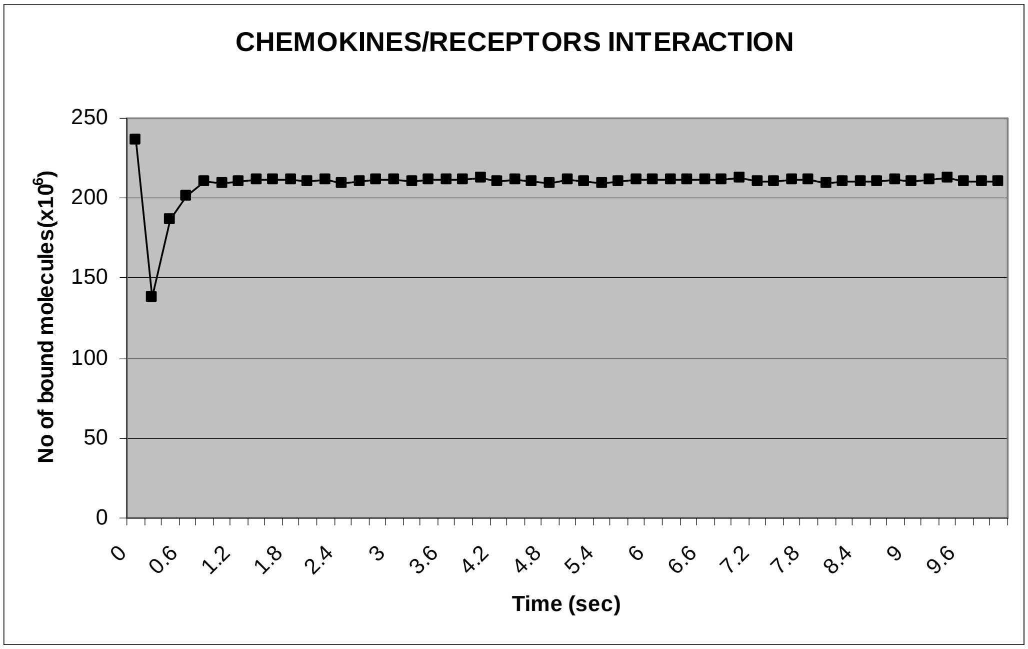


Figure 25: CHEMOKINES/RECEPTORS interaction. Number of bound molecules (x10^6) CHEMOREC_BOUND during the second phase of the lymphocyte recruitment

In Figure 25 the y-axis represents the number of CHEMOREC_BOUND molecules in millions during the second phase of the lymphocyte recruitment. The x-axis is the simulation time for which the experiment is executed (10 sec). The CHEMOKIN/RECEPTORS interaction starts with a big rise, followed by a fall and then a rise smaller than the first one. Then there is a steady straight line. The rate of disassociation of the CHEMOKIN/CHEMOREC interaction is higher than the rate of association, and that is why the number of bound molecules (around 215x10^6) is lower than the initial CHEMOKIN number of molecules (236x10^6).

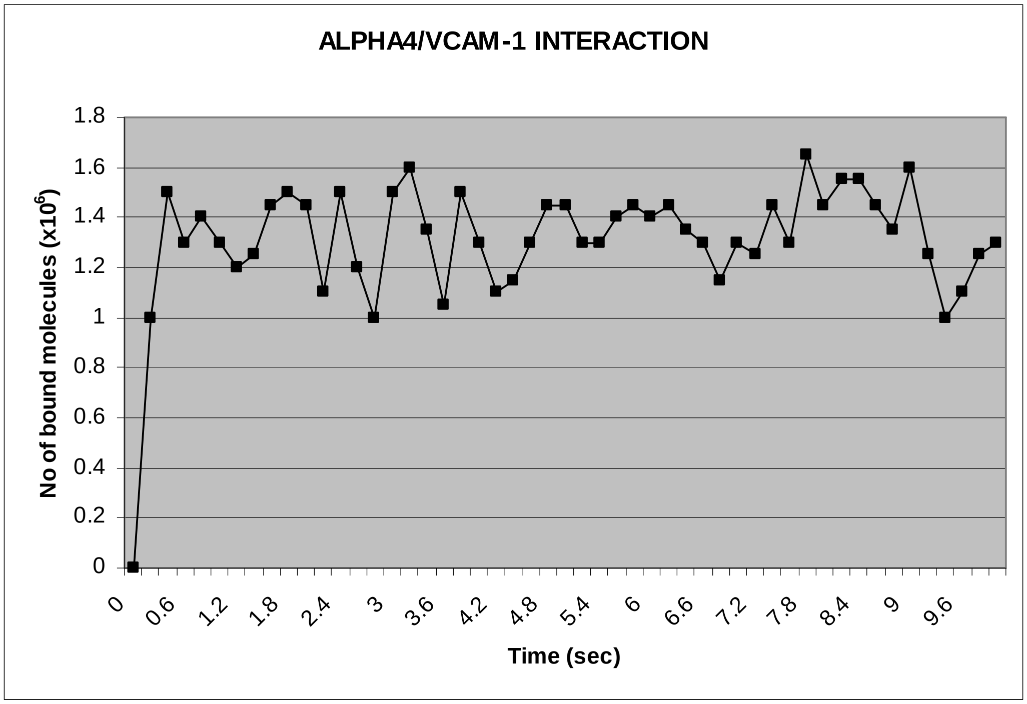


Figure 26: ALPHA4/VCAM-1 interaction. Number of bound molecules ($x10^6$) ALPHA4_BOUND/VCAM-1_BOUND during the third phase of the lymphocyte recruitment

In Figure 26 the y-axis represents the number of ALPHA4_BOUND/VCAM-1_BOUND molecules in millions during the third phase of the lymphocyte recruitment. The x-axis is the simulation time for which the experiment is executed (10 sec). Fluctuation is noticed in the ALPHA4/VCAM-1 interaction (Figure 26), with bigger range than the one in the PSGL-1/P-SELECTIN interaction. The values of ALPHA4_BOUND and VCAM-1_BOUND range from $1.00x10^6$ to $1,65x10^6$ during the fluctuation. Even though APHLA4/VCAM-1 interaction plays bigger role in the adhesion of the cell, it is still involved in the rolling/tethering phase. The forming and breakage of the bonds results this fluctuation, which is more intense than the PSGL-1/PSELECTIN interaction.

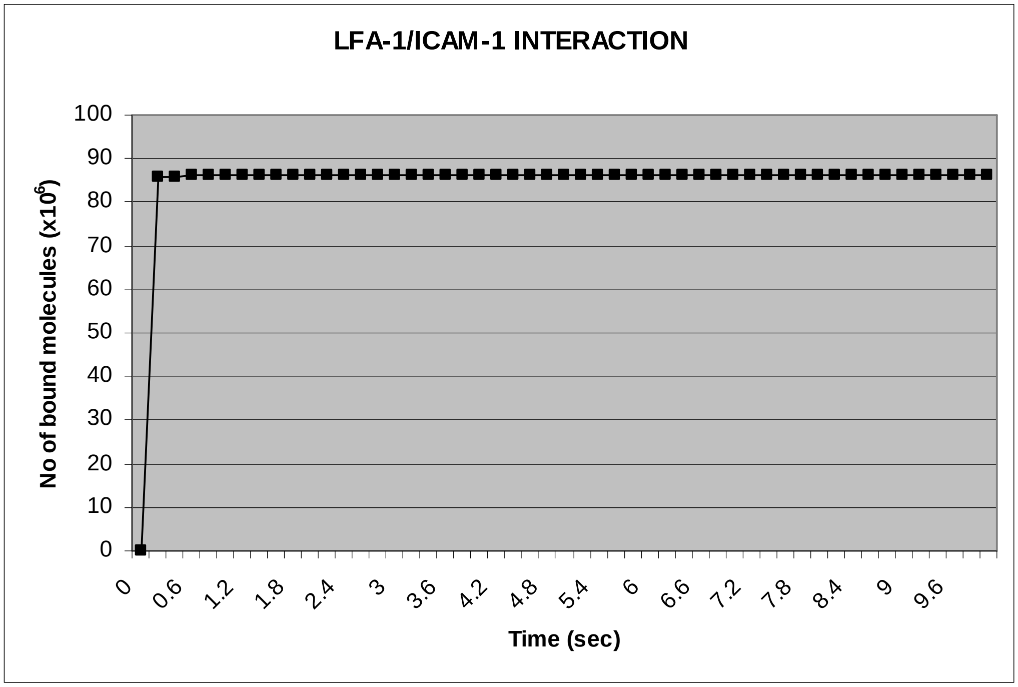


Figure 27: LFA-1/ICAM-1 interaction. Number of bound molecules (x10^6) LFA-1_BOUND/ICAM-1_BOUND during the fourth phase of the lymphocyte recruitment

In Figure 27 the y-axis represents the number of LFA-1_BOUND/ICAM-1_BOUND molecules in millions during the fourth phase of the lymphocyte recruitment. The x-axis is the simulation time for which the experiment is executed (10 sec). The graph of the LFA-1/ICAM-1 (Figure 27) starts with big rise and then a steady behaviour. High levels of LFA-1_BOUND and ICAM-1_BOUND molecules denote that there is high propensity of firm adhesion of the cell. The steady behaviour is explained by the fact that once the cell is firmly adhered then it remains there until the final diapedesis.

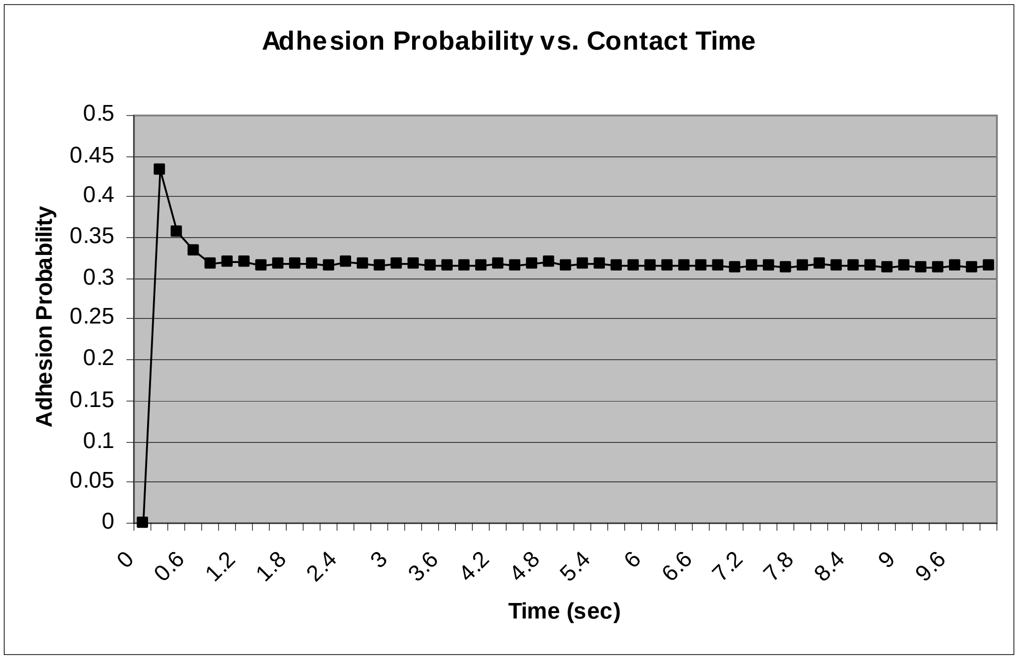


Figure 28: The adhesion probability vs. contact time

In Figure 28 the y-axis represents the adhesion probability. The x-axis is the simulation time for which the experiment is executed (10 sec). The adhesion probability starts with a pick, 0.433, and then slowly lowering to the value of 0.320. Some slight fluctuation is observed in the range 0.312-0.320. The mice under investigation are suffering form experimental autoimmune encephalomyelitis and the 31-32% possibility of adhesion is pretty high.

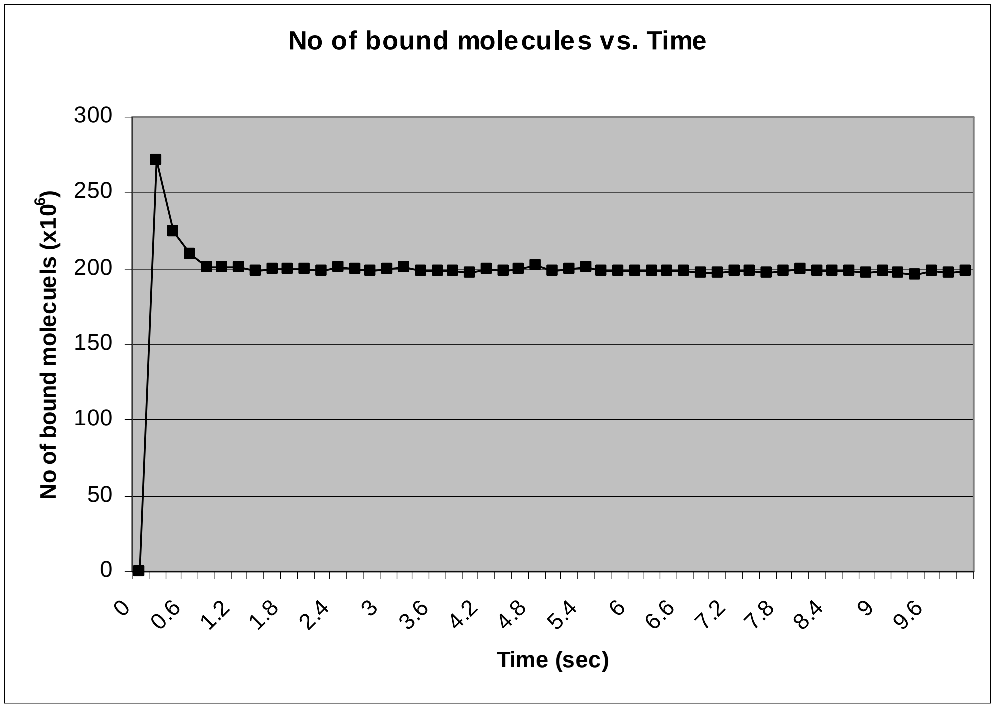


Figure 29: Number of bound molecule vs. Time

In Figure 29 y-axis represents the number of bound molecules, which means the total number of PSGL-1_BOUND, CHEMOREC_BOUND, ALPHA4_BOUND and LFA-1_BOUND. The x-axis represents the simulation time for which the experiment is executed. After a pick of $272x10^6$ bound molecules, the graph lowers to $200x10^6$. A fluctuation follows the 1.0 second ranging between the values $196x10^6$ and $202x10^6$.

5.3 Treatment under investigation

In this section we will see our endeavour to apply the treatment of interferon beta (IFN-β-1b) in our model. The graphs shown are for 3-month therapy, 6-month therapy and 12-month therapy.

Based on [25] we inserted into our model the fact that CHEMOKIN can be reduced during the treatment of a patient with interferon IFN-β-1b. In 3-month therapy with IFN-β-1b the levels of CHEMOKIN (CXCL10) is reduced from $236x10^6$ to $168x10^6$, in 6-month therapy to $163x10^6$ molecules and in 12-month therapy to $137x10^6$ molecules. The results are shown on the probability of adhesion and shown below.

Table 5: The concentrations of CHEMOKIN under therapy

Time Intervals	**CHEMOKIN (CXCL10)**
Baseline	$236x10^6$
3-month therapy	$168x10^6$
6-month therapy	$163x10^6$
12-month therapy	$137x10^6$

5.3.1 3-month therapy

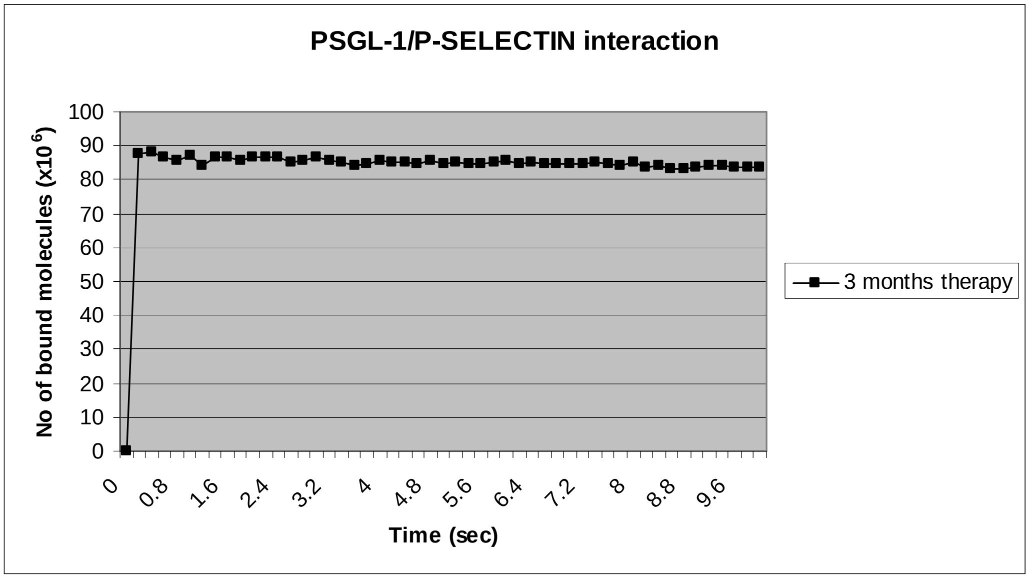


Figure 30: PSGL-1/P-SELECTIN interaction at 3-months therapy. Number of bound molecules (x10^6) PSGL-1_BOUND/P-SELECTIN_BOUND during the first phase of lymphocyte recruitment (PSGL-1/P-SELECTIN interaction) at 3 months therapy with interferon IFN-β-1b.

In Figure 30 the y-axis represents the number of PSGL-1_BOUND/P-SELECTIN_BOUND molecules in millions during the first phase of the lymphocyte recruitment at 3 months therapy. The x-axis is the simulation time for which the experiment is executed (10 sec). The PSGL-1/P-SELECTIN interaction in Figure 30 shows a big rise at the beginning of the graph. The concentrations of PSGL-1_BOUND and P-SELECTIN_BOUND reach high levels, 87.8x10^6 at the beginning and 88.0x10^6 later on. A fluctuation is observed in the next seconds at which values range from $83.1x10^6$ to $88.0x10^6$. This is due to the fact that bonds are created and break during the tethering and rolling of the cell. The results of the 3 months therapy show a decrease in PSGL-1_BOUND and P-SELECTIN_BOUND as in the baseline simulation the values of these bound molecules ranged from $84.3x10^6$ to $87.9x10^6$. This shows the contribution of interferon IFN-β-1b to the treatment.

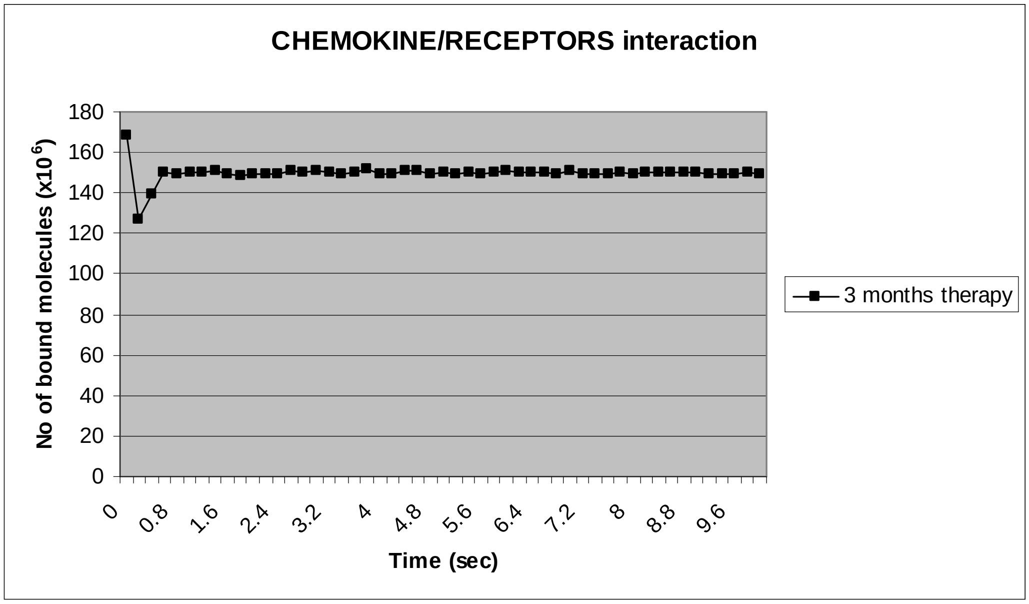


Figure 31: CHEMOKINE/RECEPTORS interaction at 3 months thepary. Number of bound molecules ($x10^6$) CHEMOREC_BOUND during the second phase of lymphocyte recruitment (CHEMOKINE/RECEPTORS interaction) at 3 months therapy with interferon IFN-β-1b.

In Figure 31 the y-axis represents the number of CHEMOREC_BOUND molecules in millions during the second phase of the lymphocyte recruitment at 3 months therapy. The x-axis is the simulation time for which the experiment is executed (10 sec). The graph starts with a pick at $168x10^6$ molecules followed by a low value of $127x10^6$. The graph proceeds with a rise, leading to a relative steady values ranging from $148x10^6$ molecules to $151x10^6$. We observe that the behaviour of CHEMOREC_BOUND is the same as the one in the baseline experiments, having though lower values as the initial value in this case is reduced to $168x10^6$.

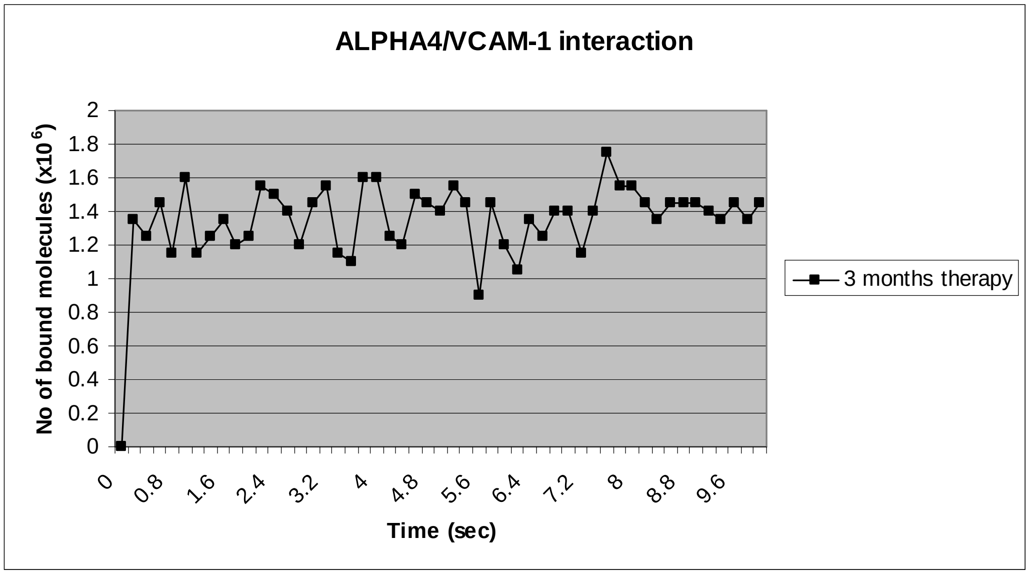


Figure 32: ALPHA4/VCAM-1 interaction at 3-month therapy. Number of bound molecules (x10^6) ALPHA4_BOUND/VCAM-1_BOUND during the third phase of lymphocyte recruitment (ALPHA4/VCAM-1 interaction) at 3 months therapy with interferon IFN-β-1b.

In Figure 32 the y-axis represents the number of ALPHA4_BOUND/VCAM-1_BOUND molecules in millions during the third phase of the lymphocyte recruitment at 3-month therapy. The x-axis is the simulation time for which the experiment is executed (10 sec). We observe that the concentration of ALPHA4_BOUND and VCAM-1_BOUND rises at the beginning of the graph, followed by a fluctuation ranging from 0.90x10^6 to 1.75x10^6. Comparing with Figure 26, the baseline experiment, we observe that the ALPHA4_BOUND and VCAM-1_BOUND molecules are in wider range of values than the baseline ones. Baseline results were between 1.00x10^6 and 1.65x10^6.

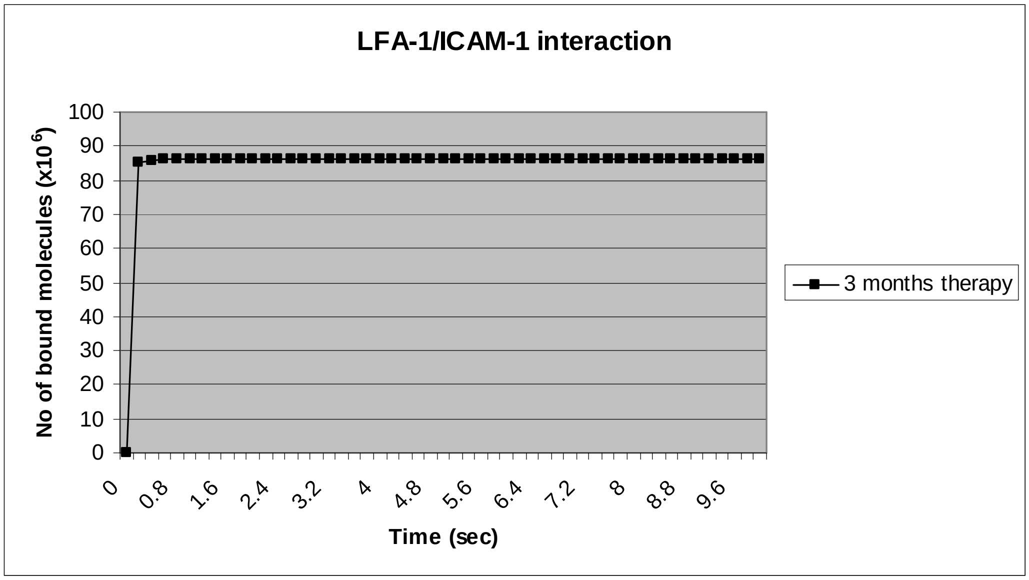


Figure 33: LFA-1/ICAM-1 interaction at 3 months therapy. Number of bound molecules ($x10^6$) LFA-1_BOUND/ICAM-1_BOUND during the first phase of lymphocyte recruitment (PSGL-1/P-SELECTIN interaction) at 3 months therapy with interferon IFN-β-1b.

In Figure 33 the y-axis represents the number of LFA-1_BOUND/ICAM-1_BOUND molecules in millions during the fourth phase of the lymphocyte recruitment at 3 months therapy. The x-axis is the simulation time for which the experiment is executed (10 sec). We observe that after the steep increase, the number of bound molecules acquires a steady value of $86x10^6$. This is the same steady value that is acquired by the baseline experiments. The firm adhesion of the cell in this phase makes the bondage of LFA-1 and ICAM-1 steady. There is no return to the free state and no breakage. The application of the interferon beta therapy does not affect this part of the phase. The results are almost exactly the same as the baseline results.

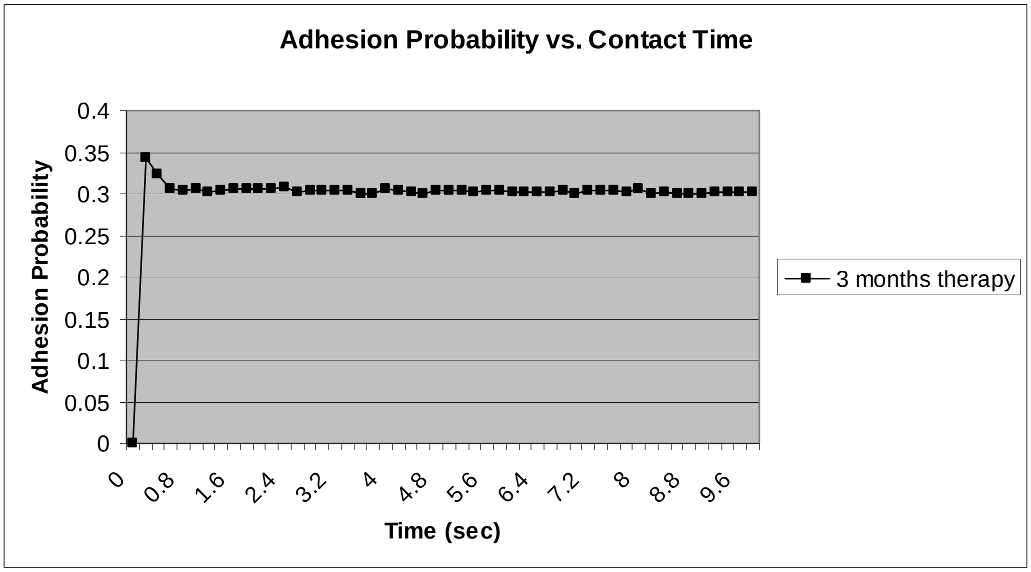


Figure 34: Adhesion probability versus contact time at 3 months therapy

In Figure 34 the y- axis represents the probability of adhesion and the x-axis the simulation time (10sec). The adhesion probability makes a big rise in the first half second of the simulation reaching the value of 0.343 and then lowers to the value of 0.305. The value then fluctuates in the range of 0.301 and 0.307. There is a small decrease in the probability of adhesion compared to the baseline simulation. In the latter case the probability was fluctuating in the range 0.312-0.320. There is a contribution of interferon IFN-β-1b to the decrease in the probability of adhesion.

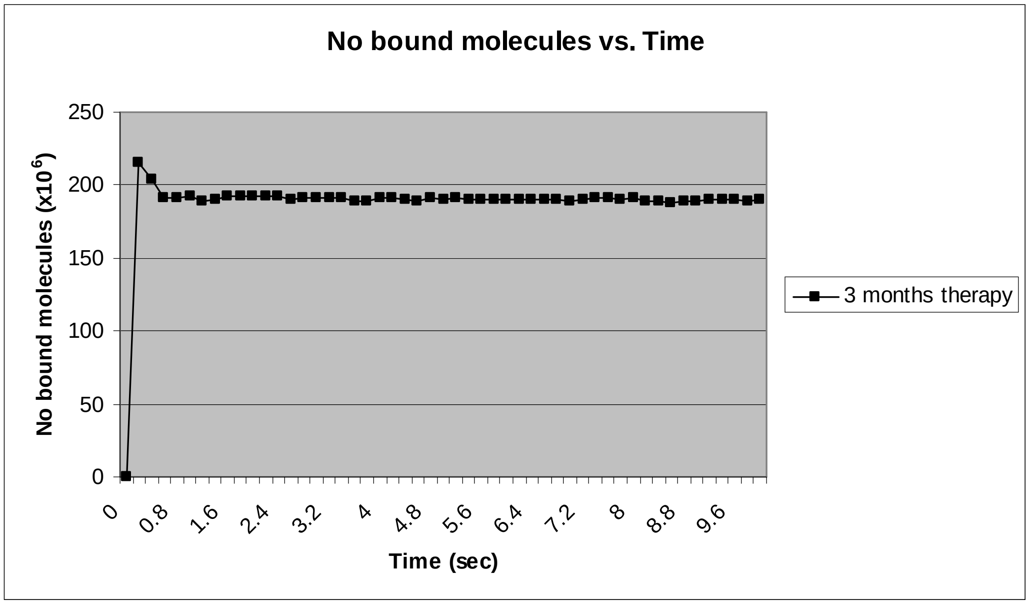


Figure 35: No of bound molecules vs. Time at 3 months therapy

In Figure 35, y-axis represents the total number of bound molecules at 3 months therapy. This means the sum of PSGL-1_BOUND, CHEMOREC_BOUND, ALPHA4_BOUND, LFA-1_BOUND. The x-axis represents the simulation time for which the experiment is executed (10 seconds). After a steep rise to 215x106 molecules the graph lowers to 192x10^6. After the first second the value is fluctuating between the values 188x10^6 to 193x10^6. Comparing these results with the baseline results we observe a decrease in bound molecules. The baseline results in Figure 29 shows a fluctuation between the values 196x10^6 and 202x10^6. This is a contribution of the interferon beta that reduces the number of bound molecules by almost 1%.

5.3.2 6-month therapy

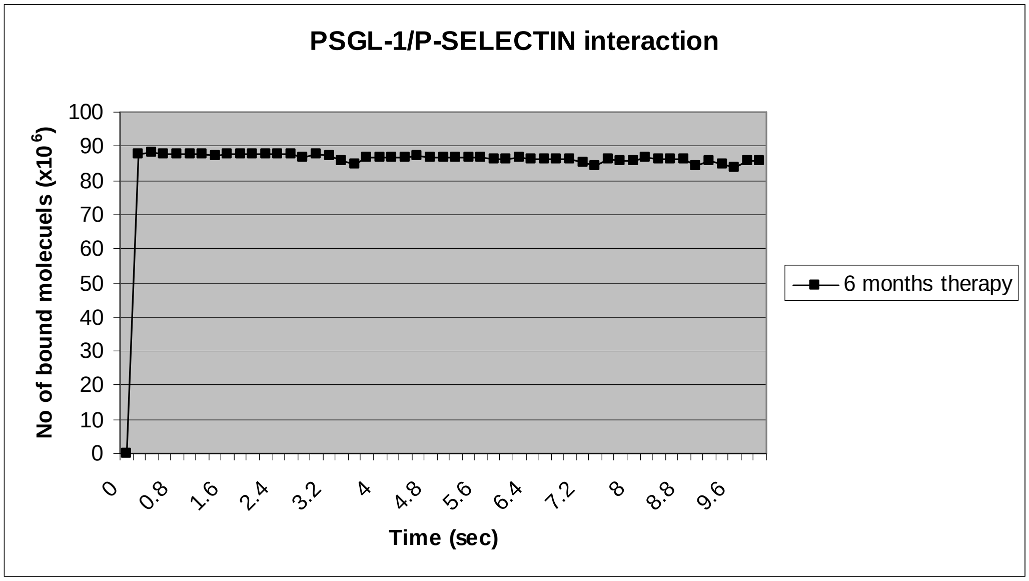


Figure 36: PSGL-1/P-SELECTIN interaction at 6 months therapy**.** Number of bound molecules (x10^6) PSGL-1_BOUND/P-SELECTIN_BOUND during the first phase of lymphocyte recruitment (PSGL-1/P-SELECTIN interaction) at 6 months therapy with interferon IFN-β-1b.

In Figure 36 the y-axis represents the number of PSGL-1_BOUND/P-SELECTIN_BOUND molecules in millions during the first phase of the lymphocyte recruitment at 6 months therapy. The initial steep rise to 87.6x10^6 molecules is followed by a fluctuation ranging between 83.7x10^6 and 88.0x10^6. The corresponding values in the baseline experiments are between 84.3x10^6 and 87.9x10^6 (fig. 24), where as the values of the 3 months therapy are between 83.1x10^6 and 88.0x10^6 (fig. 30). The results of the 3 months therapy are better than the ones in 6 months therapy. This is due to the fact that the difference of the CHEMOKINE concentrations in these two cases is very small, 168x10^6 for 3 months therapy and 163x10^6 for 6 months therapy, and our simulation obeys the exponential probabilistic distribution a factor that provides non-determinism. Comparing 6-month therapy with the baseline we observe a decrease in the concentration of PSGL-1_BOUND and P-SELECTIN_BOUND molecules by 1%.

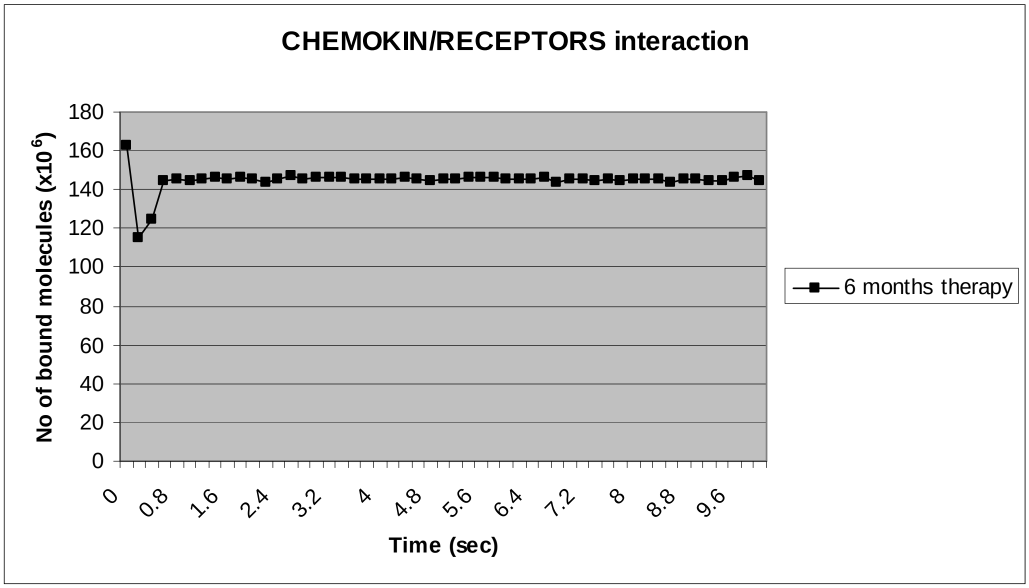


Figure 37: CHEMOKIN/RECEPTORS interaction at 6 months therapy. Number of bound molecules (x10^6) CHEMOREC_BOUND during the second phase of lymphocyte recruitment (CHEMOKINE/RECEPTORS interaction) at 6 months therapy with interferon IFN-β-1b.

In Figure 37 the y-axis represents the number of CHEMOREC_BOUND molecules in millions during the second phase of the lymphocyte recruitment at 6 months therapy. CHEMOREC_BOUND begins in high concentration, 163x10^6, then lowers to 115x10^6 and up again to 144x10^6. Fluctuation is observed after the 0.6 second of simulation with the values to range between 144x10^6 and 147x10^6.

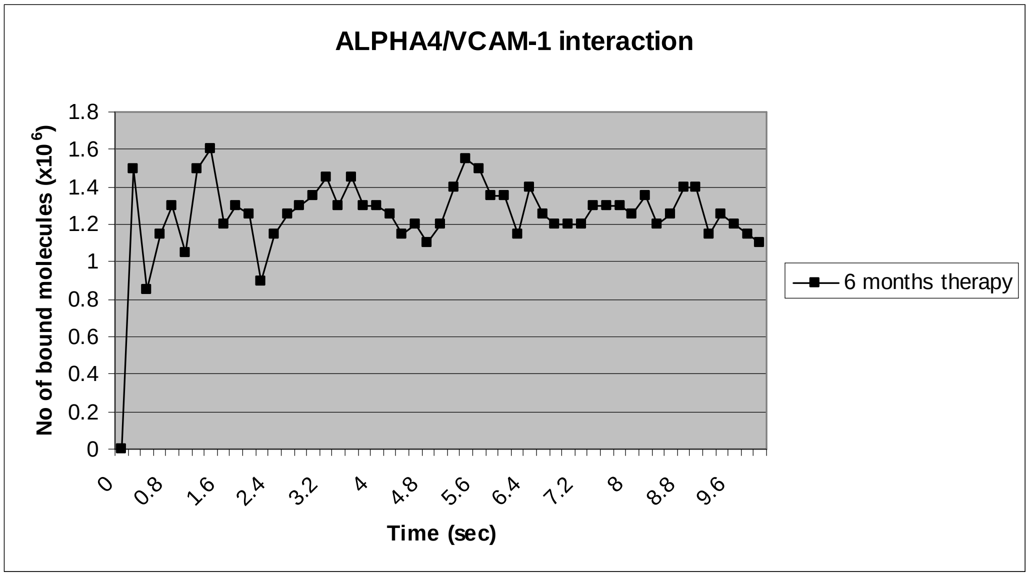


Figure 38: ALPHA4/VCAM-1 interaction at 6 months therapy. Number of bound molecules (x10^6) ALPHA4_BOUND/VCAM-1_BOUND during the third phase of lymphocyte recruitment (ALPHA4/VCAM-1 interaction) at 6 months therapy with interferon IFN-β-1b.

In Figure 38 the y-axis represents the number of ALPHA4_BOUND/VCAM-1_BOUND molecules in millions during the third phase of the lymphocyte recruitment at 6 months therapy. There is a big rise at the beginning of the graph followed by a wide-range fluctuation. The values lay between the values of 0.85x10^6 and 1.60x10^6. There is a small reduce in the values compared to the ones in the baseline experiments and the 3 months therapy. The baseline results were between 1.00x10^6 and 1.65x10^6 and the 3 months therapy results were between 0.90x10^6 and 1.75x10^6. There is a reduction of 1% from the 3 months results and 0.85% reduction from the baseline results.

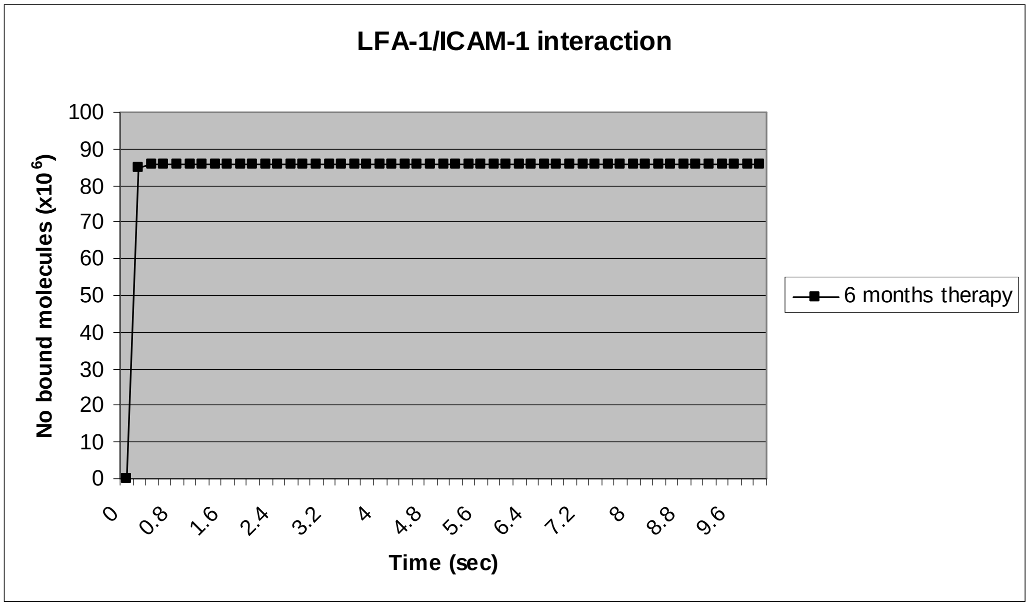


Figure 39: LFA-1/ICAM-1 interaction at 6 months therapy. Number of bound molecules (x10^6) LFA-1_BOUND/ICAM-1_BOUND during the first phase of lymphocyte recruitment (PSGL-1/P-SELECTIN interaction) at 6 months therapy with interferon IFN-β-1b.

In Figure 39 the y-axis represents the number of LFA-1_BOUND/ICAM-1_BOUND molecules in millions during the third phase of the lymphocyte recruitment at 6 months therapy. The value of bound molecules increased rapidly in the first 0.2 seconds to 85.1x106 and in 0.4 seconds to 86.0x106. From 0.4 seconds on the value stays the same. This denotes that the cell has been adhered and is ready for diapedesis. The results in this graph are the same as the graphs in Figure 27 and Figure 33 for the baseline results and the 3 months therapy respectively.

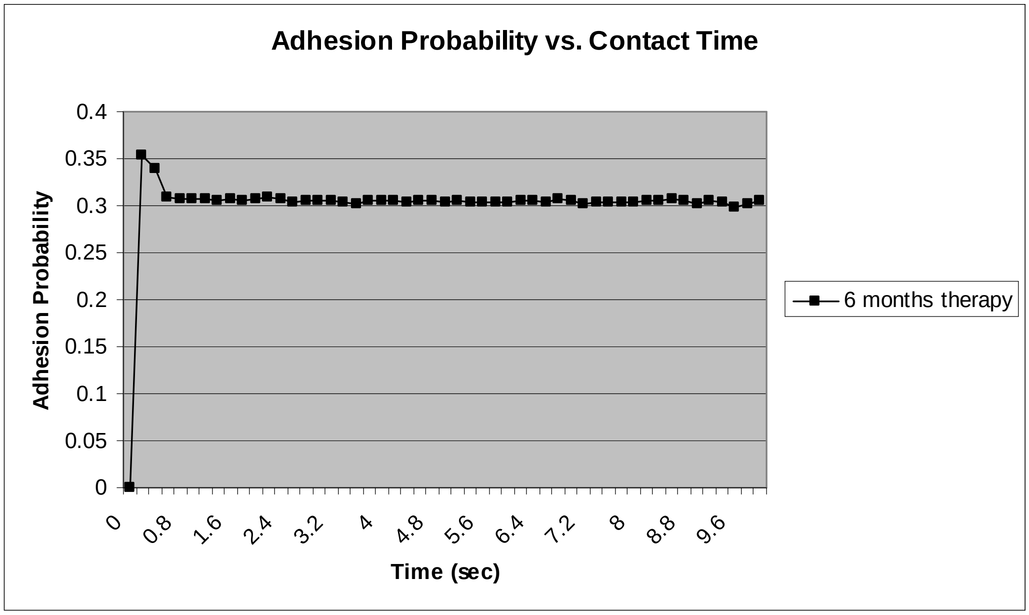


Figure 40: Adhesion probability versus contact time at 6 months therapy

In Figure 40 the y-axis represents the probability of adhesion and the x-axis represents the simulation time of the experiment at 6-month therapy. The graph shows an increase to 0.353 and then lowers to 0.308. There is a fluctuation from time 0.8 seconds to 10.0 seconds ranging from 0.298 to 0.307. The corresponding results of the baseline results are 0.312 to 0.320 and the 3-month therapy results are 0.301 to 0.307. This shows a 0.95% decrease comparing with the baseline and 0.99% decrease comparing with the 3-month therapy. It is optimistic to see a small improvement in the probability of adhesion using IFN-β-1b.

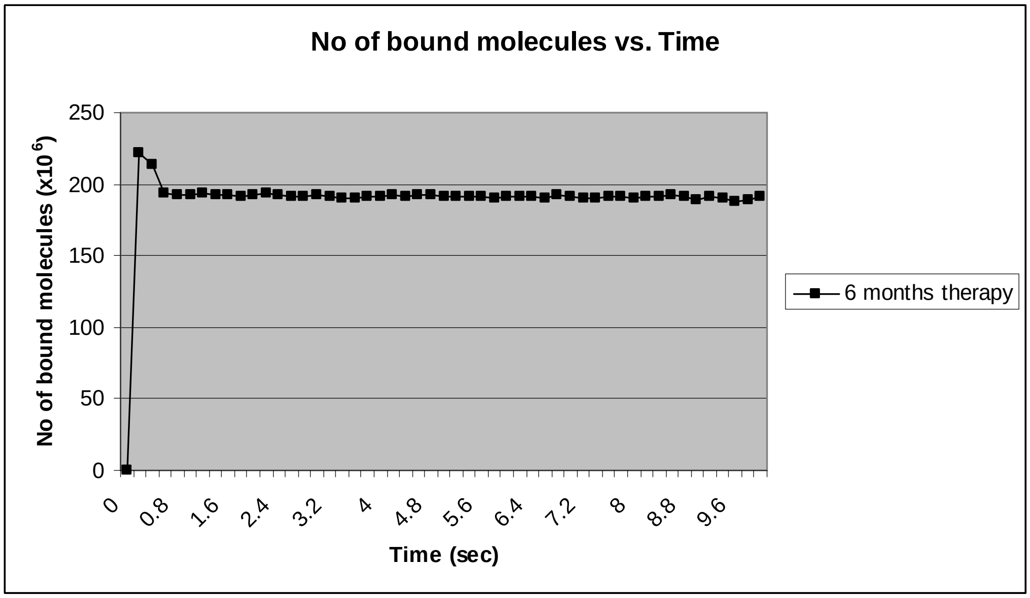


Figure 41: No of bound molecules vs. Time at 6 months therapy

In Figure 41 the y-axis represents the number of bound molecules and the x-axis represents the simulation time of the experiment at 6-month therapy. The graph shows an increase to 222x10^6 and then lowers to 194x10^6. The small fluctuation that follows is caused by the forming and breakage of the bonds between ligands and integrins. The fluctuation ranges between 187x10^6 to 194x10^6 bound molecules. The corresponding ranges in the baseline (fig. 29) and the 3 months therapy (Fig.35) are 196x10^6 to 202x10^6 and 188x10^6 to 193x10^6, respectively. Comparing with the baseline we have 0.96% decrease. Comparing with the 3-month therapy there is not that much difference.

5.3.3 12-month therapy

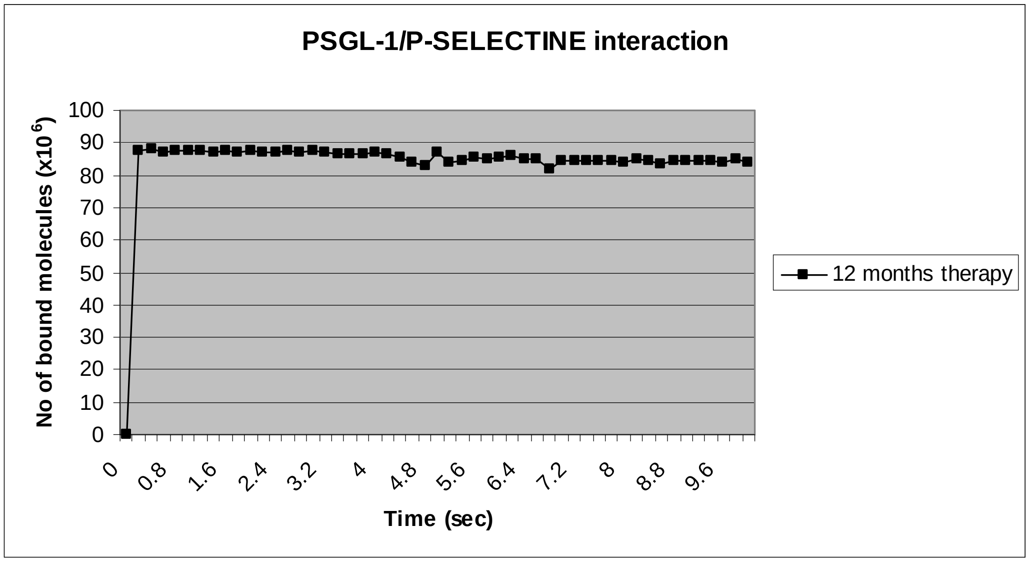


Figure 42: PSGL-1/P-SELECTIN interaction at 12 months therapy**.** Number of bound molecules (x106) PSGL-1_BOUND/P-SELECTIN_BOUND during the first phase of lymphocyte recruitment (PSGL-1/P-SELECTIN interaction) at 12-month therapy with interferon IFN-β-1b.

In Figure 42 the y-axis represents the number of PSGL-1_BOUND/P-SELECTIN_BOUND molecules in millions during the first phase of the lymphocyte recruitment at 12 months therapy. The steep increase of the bound molecules at the beginning of the simulation is followed by a fluctuation due to the forming and breakage of bonds between PSGL-1 and P-SELECTIN. Due to the high rate of association and the low rate of disassociation we have the big increase at first. The fluctuation later lies between the values of 83.3x10^6 and 87.9x10^6. This is lower than the baseline results and also lower than the 6-month therapy results. The 3-month therapy results are almost equivalent. The corresponding results of the baseline range from 84.3x10^6 to 87.9x10^6, the 3-month therapy range from 83.1x10^6 to 88.0x10^6 and the 6 month therapy results lay between 83.7x10^6 and 88.0x10^6.

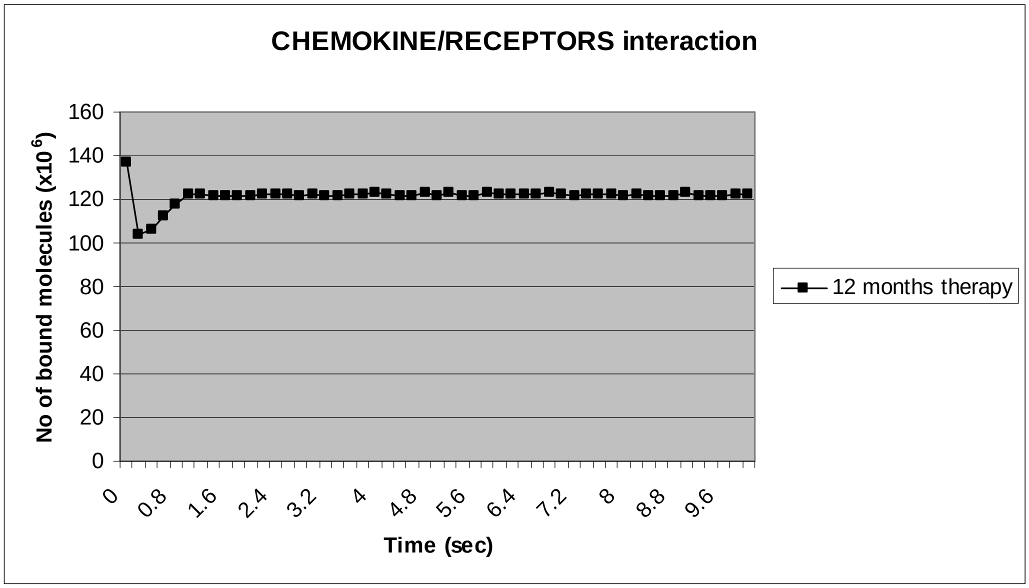


Figure 43: CHEMOKINE/RECEPTORS interaction at 12 months therapy. Number of bound molecules (x10^6) CHEMOREC_BOUND during the second phase of lymphocyte recruitment (CHEMOKINE/RECEPTORS interaction) at 12 months therapy with interferon IFN-β-1b.

In Figure 43 the y-axis represents the number of CHEMOREC_BOUND molecules in millions during the second phase of the lymphocyte recruitment at 12 months therapy. The concentration of CHEMOREC_BOUND starts with a high value of 137x10^6 then lowers to 104x10^6 and after that it rises to 122x10^6. This is followed by an almost steady value that ranges between 121x10^6 and 123x10^6. The behaviour of CHEMOREC_BOUND is almost the same as the one in the baseline and in the 3-month therapy. The CHEMOREC_BOUND does not stay in very high concentrations as the rate of association (RA_C) is lower than the rate of disassociation (RD_C).

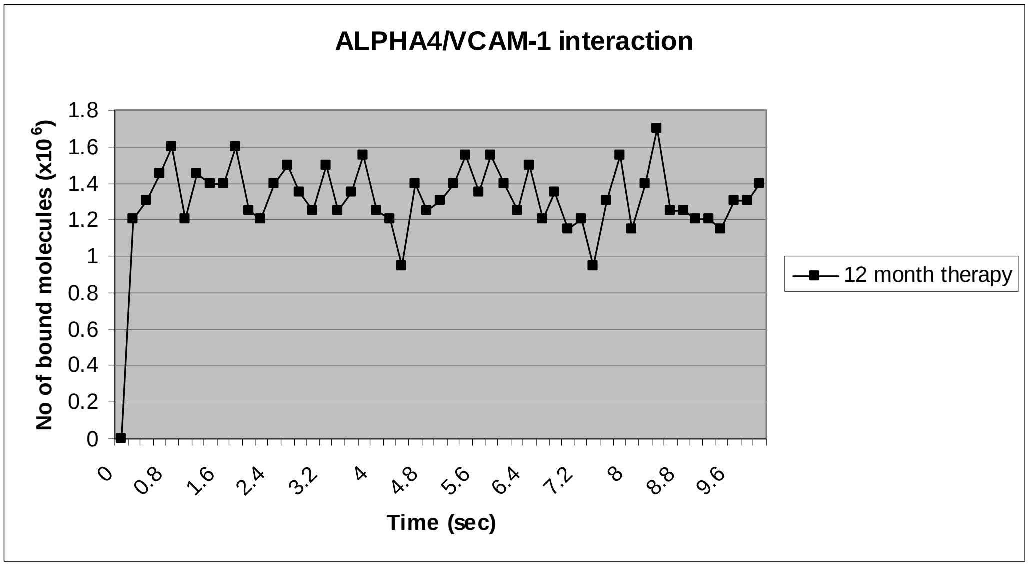


Figure 44: ALPHA4/VCAM-1 interaction at 12 months therapy. Number of bound molecules ($x10^6$) ALPHA4_BOUND/VCAM-1_BOUND during the third phase of lymphocyte recruitment (ALPHA4/VCAM-1 interaction) at 12-month therapy with interferon IFN-β-1b.

In Figure 44 the y-axis represents the number of ALPHA4_BOUND/VCAM-1_BOUND molecules in millions during the third phase of the lymphocyte recruitment at 12 months therapy. We observe an initial rise to the $1.2x10^6$ and then a wide-range fluctuation that has minimum value of $0.95x10^6$ and a maximum of $1.70x10^6$. The wide-range fluctuation is due to the bondage of the two molecules and the possible breakage that follows. ALPHA4 integrin and VCAM ligand play role in the rolling phase but also in the adhesion of the cell. Possible adhesion is not firm so there might be a disassociation from the endothelium. This explains the big increases and decreases in the concentrations of ALPHA4_BOUND and VCAM-1_BOUND. The corresponding values of baseline range between $1.00x10^6$ and $1.65x10^6$. The results from the 3-month therapy range between $0.90x10^6$ and $1.75x10^6$ and the 6-month therapy results range from $0.85x10^6$ and $1.60x10^6$. The differences are not that huge to extract conclusions. There are times that 6 months therapy gives better results than the 12-month therapy, but that is not always the case.

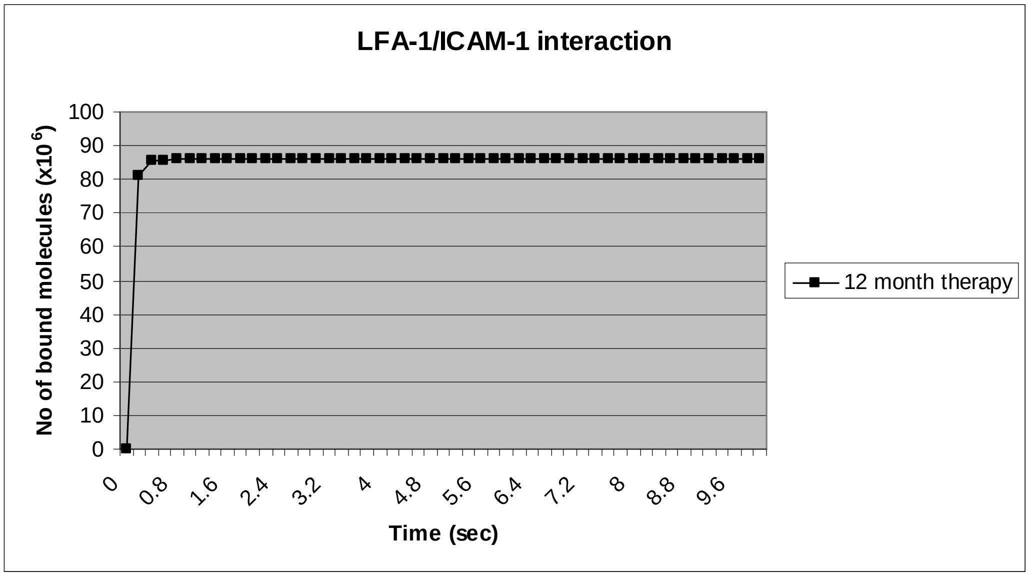


Figure 45: LFA-1/ICAM-1 interaction at 12-month therapy. Number of bound molecules (x10^6) LFA-1_BOUND/ICAM-1_BOUND during the first phase of lymphocyte recruitment (PSGL-1/P-SELECTIN interaction) at 12-months therapy with interferon IFN-β-1b.

In Figure 45 the y-axis represents the number of LFA-1_BOUND/ICAM-1_BOUND molecules in millions during the fourth phase of the lymphocyte recruitment at 12 months therapy. The concentrations LFA_1_BOUND and ICAM-1_BOUND show a big increase at the beginning of the simulation reaching the value of 81.1x10^6. Slowly they increase to the top value of 86.0x10^6, where they stay until the termination of the simulation. The high values of LFA-1_BOUND and ICAM-1_BOUND mean that the possibility of firm adhesion is high. The corresponding graphs of the baseline simulation, the 3-month therapy and the 6-month therapy show a similar behaviour with the top values reaching 86.0x10^6.

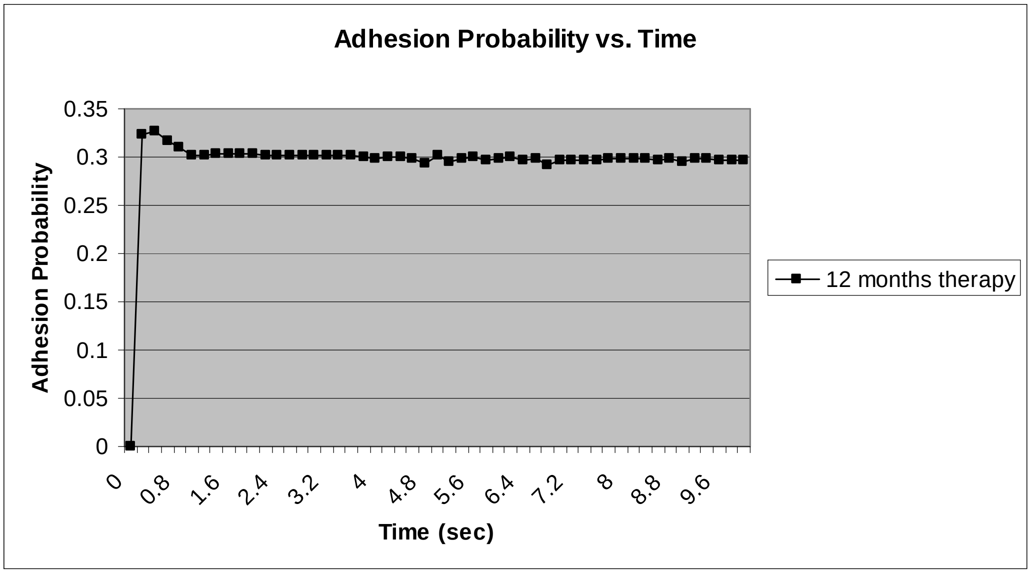


Figure 46: Adhesion probability versus contact time at 12 months therapy

In Figure 46 the y-axis represents the adhesion probability and the x-axis represents the simulation time. The adhesion probability rises at the value of 0.324 initially and then rises to the value 0.327. Until the first second of the simulation the probability lowers to the value of 0.302. The big rise at the beginning of the simulation may be considered as the worm up of the system, and later the system is reaching a steady state. From 1.0 second and on the probability shows a small fluctuation in the range of 0.292 and 0.302. Comparing this with the behaviour of adhesion probability in the baseline experiment, in the 3-month therapy and the 6-month therapy, we observe a decrease that is due to the treatment with interferon beta. The corresponding values of the baseline are between 0.312 and 0.320, the 3-month therapy results are between 0.301 and 0.307 and the 6-month therapy results are between 0.298 and 0.307. There is 2-3% decrease from the baseline, 1% decrease from the 3-month therapy and 0.6% decrease from the 6-month therapy.

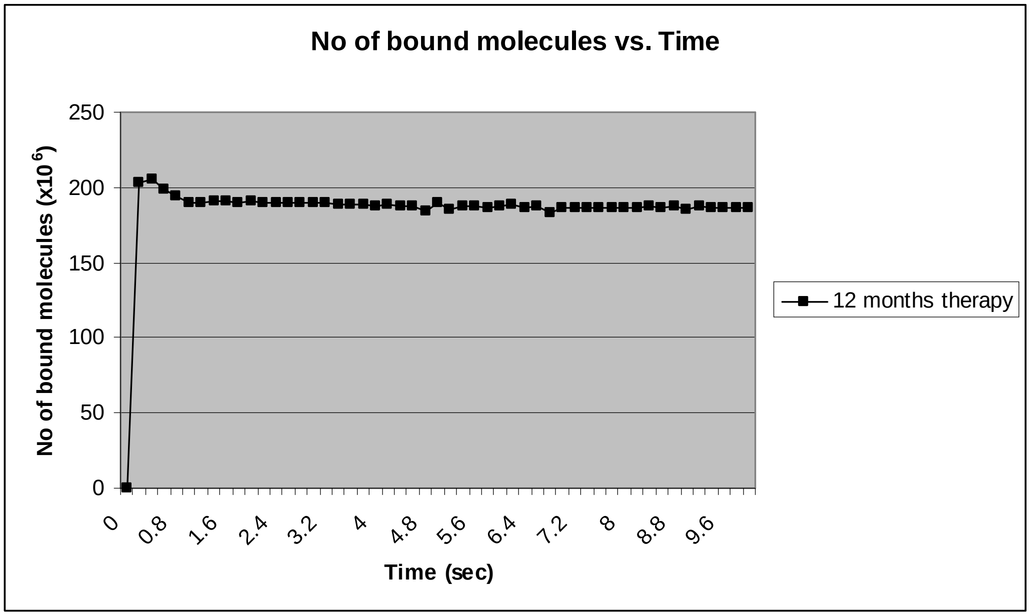


Figure 47: No of bound molecules vs. Time at 12 months therapy

In Figure 46 the y-axis represents the total number of bound molecules in millions and the x-axis represents the simulation time. By total number of bound molecules we mean the sum of PSGL-1_BOUND, CHEMOREC_BOUND, ALPHA4_BOUND and LFA-1_BOUND. We observe a big increase of the bound molecules in 0.2 seconds at the value of 203x10^6, and after a small pick of 206x10^6 there is a downward tendency to 190x10^6. The values from second 1.0 till the end of the simulation are ranging from 184x10^6 to 190x10^6. During the first second the system warms up, that is why we have the pick at the beginning of the graph. As in most of the graphs this is shown consistently and is followed by lower values that are more representative of the phenomenon. Comparing these values with the baseline results, the 3-month results and the 6-month results, we observe a decrease in the number of bound molecules as the therapy with interferon beta is in process. The corresponding results in the steady phase of the simulation are for the baseline 196x10^6 to 202x10^6, for the 3-month therapy 188x10^6 to 193x10^6 and for the 6-month therapy 187x10^6 to 194x10^6. There is a 6% decrease compared to the baseline, 2% decrease compared to the 3-months therapy and 1.6% decrease compared to the 6-month therapy.

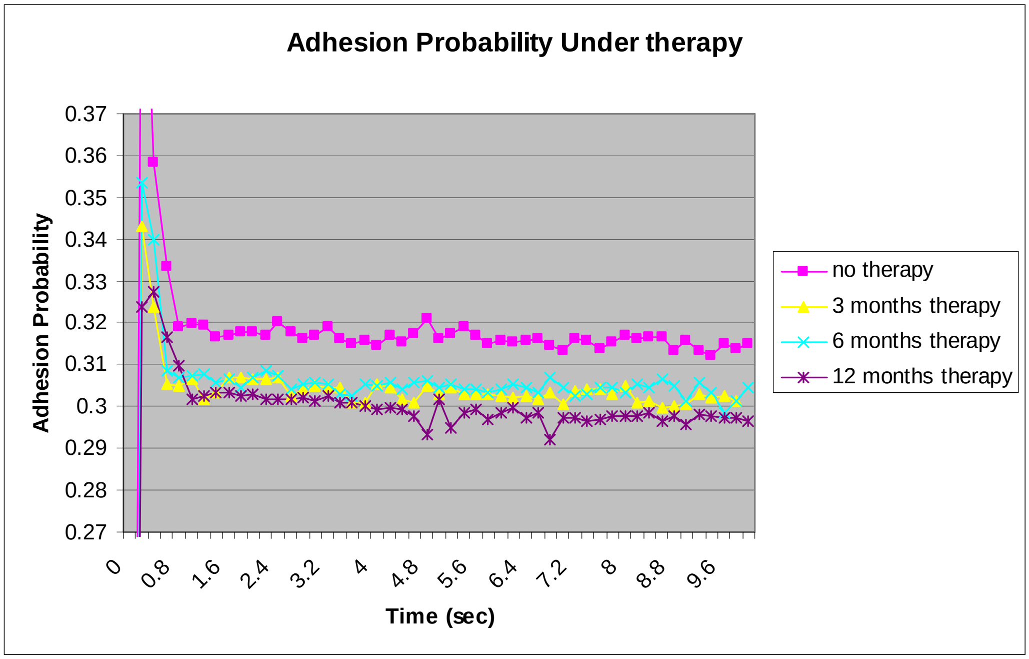


Figure 48: Adhesion probability versus contact time for all times of therapy.

Figure 48 is a visual representation of the comparison of adhesion probability in all cases of therapy and baseline. The y-axis represents the adhesion probability and the x-axis represents the simulation time in seconds. During the whole time of simulation the baseline (marked with the pink line and the square points) has greater adhesion probability than all the other cases. The 3-month therapy with interferon beta has optimistic results showing a decrease of the adhesion probability of 1.2% compared to the baseline. The 6 month therapy gives almost the same results as the 3-month therapy. In some cases the 3-month therapy gives better results. This is due to the fact that the difference of the concentration of CHEMOKIN in 3-months therapy and 6 months therapy is very small ($168x10^6$ and $163x10^6$ respectively) and the fact that probabilistic distribution is used in the model. The 12-month therapy gives in the best case 2.9% decrease in the probability of adhesion compared to the baseline results. Comparing the 12- month therapy to the 6-month and the 3-month therapy there is an improvement of 1,3%.

Generally it is noticed that the reduction of CHEMOKIN by IFN-β-1b has a downward effect on the adhesion probability. This is an encouraging fact to the treatment of multiple sclerosis, giving hope and relief to the patients of the neurological disorder.

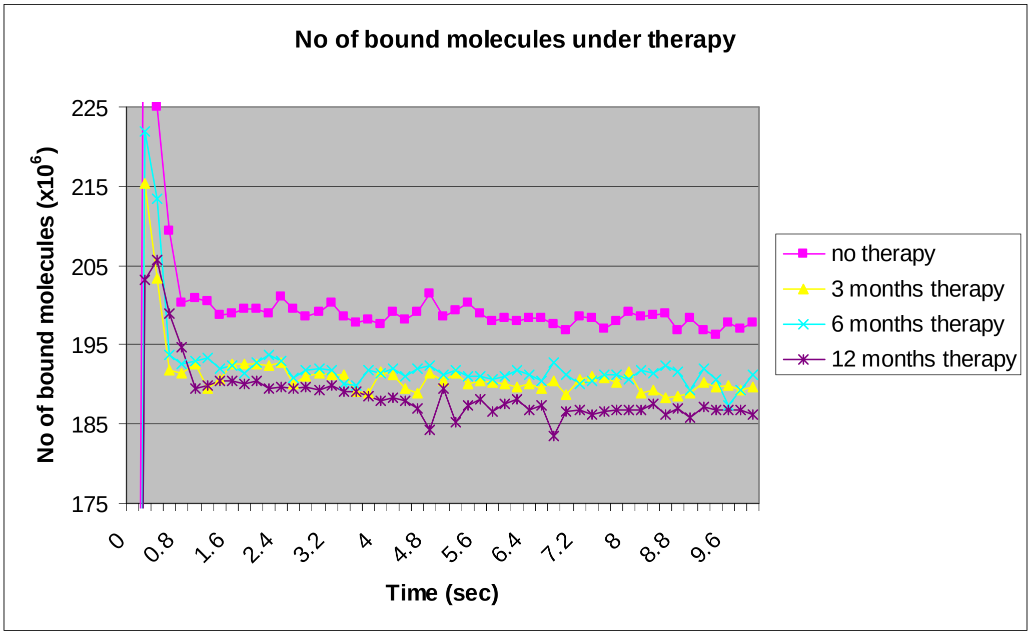


Figure 49: Number of bound molecules under the therapy with IFN-β-1b at various time intervals.

The number of bound molecules (Figure 49) is also reduced by the therapy with interferon IFN-β-1b. From 202x10^6 the bound molecules can become 184x10^6 in the 12 month therapy, an 8.91% reduction; a fact that is very optimistic for the cure of multiple sclerosis. We take into consideration the steady phase of the graph after the first second where there is a pick. Like in the probability of adhesion there is a reduction in the number of bound molecules during the 3-month therapy. Comparing the baseline with the 3-month therapy there is a decrease from 202x10^6 to 191x10^6 bound molecules at the 4.8 second, a 5% reduction. The difference in the 3 months therapy with the 6 month therapy is small, not to say that 3 months therapy gives even better result most of the times. The 12 month therapy gives a reduction to 184x10^6, which is a 3.6% decrease from the 3 month and the 6 month therapy.

Chapter 6

Conclusions and future work

The fact that in silico experiments can contribute to predictive and preventive medicine is a huge step forward to the treatment of diseases that can not be examined with in vitro or in vivo experiments. Multiple Sclerosis is such a case. The testing of drugs under certain conditions with computational simulations makes things easier to biologist and doctors. A drug can be tested under different input data and track its behaviour without any risk of living organism. One can estimate the indicated dose and dosage of a drug that gives the best results under given input.

The fact that Mobius can give us a probabilistic distribution of a certain activity to happen indicates a more proper way to look things in simulations of real life experiments. The non-deterministic nature of Stochastic Petri Nets is a closer approach to real life phenomena. Laws, like the mass law, take for granted that some phenomenon will happen deterministically just because the mass of two substances reside in the same space at a certain time. That is not how nature actually works. There is a possibility of two substances to interact and the interaction has a time delay that is not always unnoticed. The mathematical ODE approach is a deterministic way to examine things. Process Algebra and pi-calculus is a mathematical approach that can add stochasticity. Stochastic pi-calculus combines non-determinism with mathematical equations. Stochastic Petri Nets give a visual presentation of the model that is more understandable to people and stochasticity that is vital to Systems Biology.

The case study was curried out with set up data from mice suffering from autoimmune encephalomyelitis. By applying interferon beta there is a reduction in probability of adhesion by almost 3%. This is an optimistic result that gives hopes to people suffering from multiple sclerosis. The improvement is done along a 12 month interval of time but still is worth doing it. Interferon beta is not given to all patients as not all of them respond to this treatment. The cause of multiple sclerosis is not yet found. The recruitment of lymphocytes is one of the causes but that is not always the case. Other possible causes are virus infection and psychological reasons. It is still bothering the researchers the fact that women are more infected than men in the ratio 2:1. Multiple Sclerosis affects especially young people in the age range 20-40. People emigrating for studies abroad show higher risk.

The tool we used, Mobius is well-built, user friendly and easy to install and use. I would definitely recommend biologists to use it and run their simulations on it. Mobius has free academic licence and a good technical support. One can enrol the mailing list of the support group and be informed about other users' problems or send questions on problems he encounters. Trying to install other tools like PEP tool or Cell Illustrator, several problems occurred that prevent me from using them. As far as stochastic Petri Nets is concerned, Mobius is best tool to deal with this formalism.

Stochastic Petri Nets is a formalism that can handle quantitative analysis in a very good way. One can use big concentrations of elements without any problem. Extended places in Mobius can provide you with the ability to extend to big markings and still work in an efficient and powerful way.

Such simulations are good insights of the autoimmune diseases that are still unexplored. Autoimmune diseases affect people at unsuspected time and do not give much hope for cure. Patients are usually under long-term medication giving some improvement but not total cure.

Bibliography

[1] Wikipedia Foundation, Inc. Process Algebra. *Wikipedia, the free encyclopedia.* [Online]. Available: http://en.wikipedia.org/wiki/Process_algebra

[2] S. Hardy and P.N. Robinllard, "Modelling and Simulation of Molecular Biology Systems Using Petri Nets: Modeling Goals of Various Approaches" *Journal of Bioinformatics and Computational Biology*, vol. 2, no. 4, 2004, pp. 619-637.

[3] W. Materi and D. S. Wishart, "Computational systems biology in drug discovery and development: methods and applications," Elsevier, 2007.

[4] J. Fisher and T.A. Henzinger, "Executable cell biology," *Nature Biotechnology*, vol. 25, num. 11, 2007.

[5] A. Richard, J.P. Comet and G. Bernot, "Formal Methods for Modeling Biological Regulatory Networks," Springer, 2006.

[6] S. Gilmore, J. Jillston, "Performance Evaluation comes to Life: Quantitative Methods applied to Biological Systems," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, issue 4, 2008.

[7] U. Alon, "Transcription Networks: Basic Concepts," in *An Introduction to Systems Biology: Design Principles of Biological Circuits*. London, UK: CRC Press, Taylor & Francis Group, 2006, pp. 5-25.

[8] A. Phillips and L. Cardelli, "Efficient, Correct Simulation of Biological Processes in the Stochastic Pi-calculus," in *Conference of Computational Methods of Systems Biology, 2007*, pp. 184-199.

[9] M. Huth and M. Ryan, *Logic in Computer Science: Modeling and Reasoning about Systems*, 2nd ed., Cambridge University Press, pp. 361-366, 2004.

[10] A. Naldi, D. Thieffry and C. Chaouiya, "Decision Diagrams for Representation and Analysis of Logical Models of Genetic Networks," in *Conference of Computational Methods of Systems Biology 2007*, LNBI 4695, Springer, pp. 233–247.

[11] H. Matsuno, A. Doi, M. Nagasaki and S. Miyano, “Hybrid Petri Net Representation of Gene Regulatory Network,” in *Pacific Symposium on Biocomputing, 2000.*

[12] M. Heiner and I. Koch, “Petri Net Based Model Validation in Systems Biology,” in *International Conference on Application and Theory of Petri Nets, 2004*, pp. 216-237.

[13] L.J. Steggles, R. Banks and A. Wipat, “Modelling and Analysing Genetic Networks: From Boolean Networks to Petri Nets,” in *Conference of Computational Methods of Systems Biology, 2006*, LNBI 4210, pp. 127-141.

[14] R. Milner, J. Parrow and D. Walker, “A calculus of Mobile Processes,” in *Information and Computation* 100(1) pp.1-40, 1992.

[15] S. Van Bakel, I. Khan, M.G. Vigliotti and J. K. Heath, “Modelling intracellular fate of FGF receptors with BioAmbients” in *Workshop of Quantitative Aspects of Programming Languages , 2008.*

[16] M. Kwiatkowska, G. Norman and D. Parker, “Quantitative Verification Techniques for Biological Processes,” *Algorithmic Bioprocesses*, Natural Computing Series, Springer, 2009.

[17] *PRISM Manual*, University of Oxford, ver. 3.3.1, 2009.

[18] N. Bonzanni, K. A. Feenstra, W. Fokkink, and E. Krepska, “What can Formal Methods bring to Systems Biology?,” in *Formal Methods 2009, Second World Congress*, pp.16-22.

[19] N. Kam et al., “Formal Modeling of C. elegans Development: A Scenario-Based Approach,” in *Conference of Computational Methods of Systems Biology, 2003*, vol. 2602 of LNCS, Springer, pp.4-20.

[20] R. Banks, L.J. Steggles, “A High-Level Petri Net Framework for Genetic Regulatory Networks”, Journal of Integrative Bioinformatics 2007.

[21] J. -P. Comet, H. Klaudel, and S. Liauzu, “Modelling Multi-valued Genetic Regulatory Networks Using High-Level Petri Nets,” in *International Conference on Application and Theory of Petri Nets 2005,*LNCS 3536, Springer, pp. 208-227.

[22] W. H. Sanders and The Board of Trustees. Mobius Manual. University of Illinois. Version 2.3, 2009. [Online] Available: http://www.mobius.illinois.edu/manual/MobiusManual.pdf

[23] Wikimedia Foundation, Inc. Multiple Sclerosis. *Wikipedia, the free encyclopedia*. [Online]. Available: http://en.wikipedia.org/wiki/Multiple_sclerosis

[24] P. Lecca, C. Priami, P. Quaglia, B. Rossi, C. Laudanna and G. Constantin.. "A stochastic Process Algebra Approach to simulation of Autoreactive Lymphocyte Recruitment," *SIMULATION*, vol.80, issue 6:273-288, 2004.

[25] A. Cucci, et al., "Pro-inflammatory cytokine and chemokine mRNA blood level in multiple sclerosis is related to treatment response and interferon-beta dose," *Journal of Neuroimmunology*, 2010.