Immune System Modeling and Analysis using Bio-PEPA

Roxana Dorina Moldovan
Communications Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
roxana.moldovan@com.utcluj.ro

Eneia Nicolae Todoran
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
eneia.todoran@cs.utcluj.ro

***Abstract*— By using membrane computing patterns and the stochastic process algebra Bio-PEPA we develop and analyze a model of the immune system response against virus attacks. We identify formal conditions expressed based on Bio-PEPA functional rates for successful immune responses.**

Keywords*— *immune system, membrane computing, process algebra, Bio-PEPA, stochastic simulation, ODE simulation

I. INTRODUCTION

The human body is continuously protected by the immune system, a complex network of biological structures and processes whose primary task is to defend us against disease. The immune system comprises cells, tissues and organs that work together to defend our body against attacks of foreign agents such as bacteria, viruses or parasites.

Viruses and bacteria can enter into the human body aiming to destroy the immunity of the tissues by means of replication and propagation. The life time of such intruders depends on the host organism where they reside. A virus that resides within a cell doesn't execute any harmful actions until it becomes a mature virus. Every virus has a maturity period; in this paper we refer to this period as the M period. After the maturity period ends, the virus starts replicating and sending copies of itself to other cells. The replication process occurs at every P units of time, where P is the propagation period. One important feature of the immune system is the capability of self-detecting the presence of intruders, being able even to neutralize viruses. Virus neutralization is the process of virus annulment, involving both a virus and an antibody. Antibodies are self-generated by the organism and their purpose is to protect the cells by destroying the detected virus particles. If the antibody is not present in a membrane then the inter-cellular communication makes it possible that the antibody to be transported from the point where it is produced to the place where it is needed.

In this paper we use the stochastic process algebra Bio-PEPA [7] to model and analyze the behavior of the immune system. We analyze the response of the immune system against virus attacks. Following [4], the structure of the system is modelled using membrane computing patterns [14] and, essentially, it is a tree-like structure. Formal conditions are identified and expressed based on Bio-PEPA functional rates for successful immune responses against virus attacks.

In terms of distributed computing we can say that the immune system is a massively concurrent environment. In this environment it is possible that sometimes the amount of the existing resources to be deficient compared to the demand for resources. This is why all the cells in our body must develop a constant activity. The immune system solves this problem by using a multilayered architecture of structural barriers, for example using physical barriers (skin or membranes), physiological ones (pH value) or its own cells and molecules that provide an adaptive response immune system.

There are various formalisms that can be used to express the behavior of distributed systems. In this paper we use membrane computing patterns to express the structure of the immune system. Also, we use the stochastic process algebra Bio-PEPA to model its behavior. Some papers [3, 11] report the usage of Bio PEPA framework in investigating the immune system, but the models presented in [3, 11] do not employ membrane computing patterns for this purpose. Biological systems can be easily modeled as concurrent systems: biochemical species may correspond to the processes that interact to each other, while the reactions may be seen as distributed actions. The structure of the system can be expressed in a compositional manner by using process algebras. A series of complex analysis can be performed in order to predict the evolution of the system, the possible inconsistencies or in order to validate the model.

The immune system acts like a complex network comprising interconnected nodes. The high complexity makes the system hard to understand and especially to analyze. Given an initial layout it is useful to know when a balance is achieved, in other words it is useful to know whether the cells are clean or are infected. In pharmaceutical terms, this indicates whether the body requires external aid offered through drugs or if the body is strong enough to withstand the virus attacks. Some of these configurations would get worse in subsequent phases of the disease (for example in cancer). Therefore the detection of such patterns in the early stages would enhance cleaning the body's cells [4].

The paper is structured as follows. Section II offers a brief overview of the membrane computing paradigm followed by a description of the Bio-PEPA tool. The following sections focus on the mapping between the membrane structure and the structure of the immune system. Section III presents a Bio-PEPA modelling approach used in order to describe the interaction process between viruses and antibodies. Within Section IV the Bio-PEPA model is analysed using both stochastic and ordinary differential equations algorithms. Section V discusses directions for future research. Section VI presents some concluding remarks.

II. PRELIMINARIES

In this section we give a brief overview of the membrane computing paradigm and a description of the Bio-PEPA tool.

A. *Membrane computing*

Membrane computing (MC) is a new branch of computer science which relies on the mapping between the distributed and parallel computing domain and the operating mode of living cells, focusing on the organizational model of the cells within the tissues. MC addresses the distributed computing models by processing multisets of objects encapsulated within membrane delimited compartments forming the membrane structure. Intuitively, a multiset is a collection in which an element may occur more than once; see, e.g., [1]. The membrane structure is illustrated in Fig.1, which is based on [13]. The systems modeled based on the membrane computing abstract ideas are called membrane systems or P systems. A comprehensive introduction to membrane computing is provided in [13].

The communication between the compartments and with the outside world has an important role within this kind of model. If each membrane is associated with a host, then a membrane comprising several similar sub-membranes can be seen as a subnet and moving on with this analogy, at a very wide scale, the skin membrane can even represent the World Wide Web. Actually, the Internet packet routing process is analogous to the message exchange within P systems [4].

The advantages of using membrane computing concepts in modeling the immune system are discussed in [2]. From the membrane computing perspective, the organism being infected is represented as a skin membrane or a P system. Membrane computing focuses on the multisets of objects encapsulated in the membrane structure. Cellular hierarchy is important in the immune system and it is modeled as a tree of membranes. For example, an organ system (grouped as a membrane) includes several organs represented as sub-membranes. The viruses and antibodies are modeled as objects in the system and their properties, such as type or lifetime are symbols of objects.

B. *Bio-PEPA*

Bio-PEPA is a process algebra framework used to model and analyze the biochemical networks [18, 7]. It is actually an improvement of PEPA (Performance Evaluation Process Algebra) [10] that was originally defined for the performance

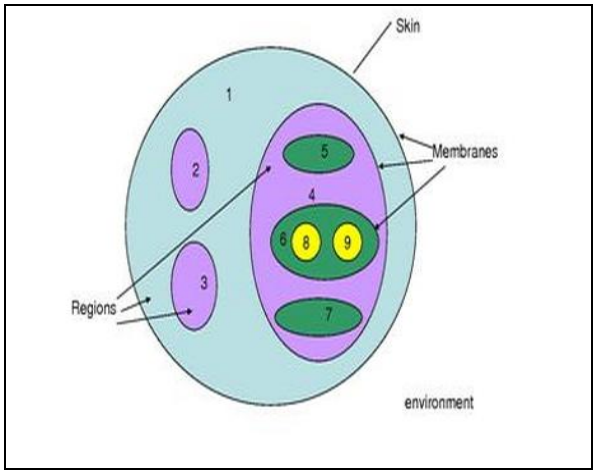

Fig.1. Membrane structure

analysis of computer systems. Bio-PEPA handles some features of biological models such as stoichiometry and general kinetic laws [7]. Stoichiometry is a branch of chemistry that deals with the relative quantities of reactants and products in chemical reactions.

A Bio-PEPA model may be seen as an intermediate formal and compositional representation of a biological system that allows the execution of a series of analyzes, including stochastic simulation, analysis based on differential equations, CTMC (continuous time Markov chain) and numerical solution for stochastic model checking using PRISM. The use of different types of analysis promotes understanding the way the system works. Functional rates are introduced into the language in order to express the general kinetic laws. Each action type is a reaction and it is associated with a functional rate [17]. Bio-PEPA also includes an operational semantics and a stochastic labeled transition based system that relies on discrete levels of concentration [7]. The representation of such discrete levels of concentration is reflected in the definition of continuous Markov chains derived from the system.

A biochemical system M is comprised of a set of compartments C representing species locations, a set of chemical species S (genes, proteins) and a set of irreversible reactions R [7]. In biochemical reactions every species has an initial concentration and this concentration can be affected by the reactions. The general form of an irreversible reaction j is illustrated in equation (1) where A_{hj} are the reactants, B_{lj} are the products, E_{vj} are the enzymes and I_{uj} are the inhibitors of the reaction. They all belong to the set S.

Enzymes and inhibitors are represented different from the reactants and products because their role is to activate or inhibit the reaction. Enzymes are molecules that accelerate chemical reactions and inhibitors are substances that reduce the rate of a chemical reaction or even prevent a chemical reaction. k_{hj} and k'_{lj} parameters are stoichiometric coefficients and express the degree in which the species participates in a reaction.

$$k_{1j}A_{1j}+\ldots+k_{nj}A_{nj} \xrightarrow{E_{1j},E_{2j},\ldots I_{1j},I_{2j}\ldots;\ f_j} k'_{1j}B_{1j}+\ldots+k'_{mj}B_{mjj} \quad (1)$$

Reaction dynamics is described by the kinetic law f_j. Reversible reactions can be seen as a pair of direct and inverse reactions. The best known kinetic law is the law of mass-action: the reaction rate is proportional to the product of the reactants' concentrations [8].

The compartments used in Bio-PEPA are static, are not actively involved in the reactions and are interpreted as simple containers. These containers are static in order to keep the simplicity of the language and to allow the representation of the majority of the biochemical networks' attributes. For example, the transport of a species from one compartment to another is modeled by introducing two different species. Translocation is abstracted by the transformation of one species into another one. When defining a Bio-PEPA system it is recommended to define all the compartments because in the system analysis phase specifying the compartments' dimension may be necessary. An example of such an analysis model is Gillespie's algorithm [17].

Reference [9] presents the syntax allowed by the version 0.1.0 of the Bio-PEPA Eclipse Plug-in and gives a brief overview of the features present within the plugin. In the following, the Bio-PEPA version 0.2.1 is going to be used for modeling the immune system response to virus attacks. Some suggestive Bio-PEPA Eclipse Plug-in syntax examples are presented in [17, 18]. The syntax differences between the Bio-PEPA symbols and the ASCII representation of these symbols within the Bio-PEPA Eclipse Plug-in are described in [9].

III. Simulating the Immune System using Bio-PEPA

In the following the membrane computing patterns and the Bio-PEPA process algebra are used to model and analyze the behavior of the immune system. The mapping between the membranes and the tree data structure the immune system relies on is presented in Fig.2. The skin membrane is associated with the whole organism and it is represented as the tree root. All other membranes reside within the skin membrane. Fig.2 illustrates four groups of cells placed within four locations. The current solution aims to exemplify the basic immune system concepts and can be further extended in order to use a higher granularity. The proposed approach considers the immune system as a hierarchy of membranes subsequently expressed and analyzed using the Bio-PEPA Eclipse Plug-in.

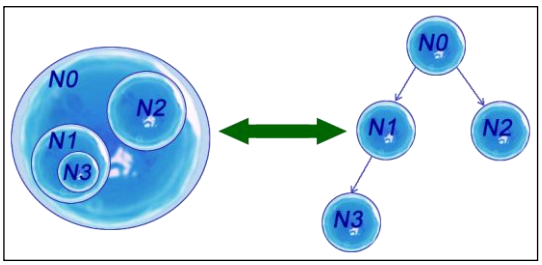

Fig.2. Immune response model architecture

```
//Locations
location N0: size=4, type=membrane;
location N1 in N0: size=2, type=compartment;
location N2 in N0: size=1, type=compartment;
location N3 in N1: size=1, type=compartment;
```

Fig.3. Bio-PEPA Eclipse Plug-in syntax for the immune system model - locations definition

Fig.3 presents the Bio-PEPA definitions of the four locations. The Bio-PEPA Eclipse Plug-in defines two types of locations: "*compartment*" and "*membrane*". The compartments used in Bio-PEPA are not actively involved in the reactions and are interpreted as simple containers. The example illustrated in Fig.3 considers all tree nodes as "compartment" locations, except for the skin membrane which is defined as a "membrane" location.

The initial state of the system assumes that a virus reaches the skin membrane and starts to replicate and propagate as shown in Fig.4. Each virus has a maturity period M and a propagation period P [4]. The viruses and the antibodies are modeled as populations implemented as Bio-PEPA species that reside within the defined locations. The four virus species managed within this approach are: *VirusPopulationN0*, *VirusPopulationN1*, *VirusPopulationN2* and *VirusPopulationN3*. These four species reside within the nodes *N0*, *N1*, *N2* and *N3* and their definitions are presented in Fig.5. Virus species may act as "reactants" or as "products" (or both), depending on their role in the biochemical reactions. For example, the *increaseVirusPopulation_N0* kinetic law is defined as a "prefixProd" Bio-PEPA semantic rule [7], where the virus species is both a product and a reactant and the level species' concentration level increases each time the reaction takes place. The stoichiometric coefficient "2" indicates that the virus replicates every time the reaction occurs.

Each virus species executes some basic actions like replication and propagation. These actions are modeled as kinetic laws. Such a kinetic law is to be defined for each of the specified actions: *increaseVirusPopulation_N0*, *sendVirusfromNotoN1* and *sendVirusfromN0toN2*. The replication is a reaction that involves only one species (e.g. VirusPopulationN0) and describes the population behavior in prefix terms [7]. In other words, the replication implies only one species and the product is represented by the same species, but with a higher concentration. For example, *increaseVirusPopulation_N0* kinetic law describes the virus replication at every M units of time and it is modeled using the functional mass-action kinetic law (fMA).

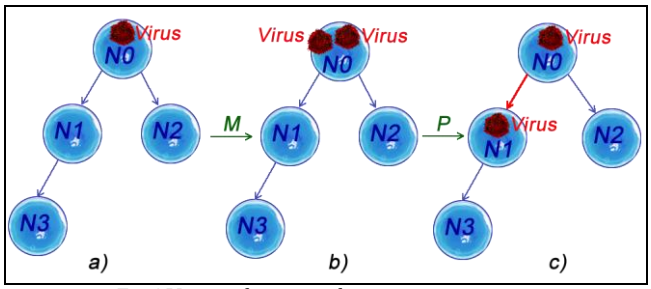

Fig.4.Virus replication and propagation processes

```
VirusPopulationN0 = (increaseVirusPopulation_N0,2) >>
VirusPopulationN0 + sendVirusfromN0toN1 <<
VirusPopulationN0 + sendVirusfromN0toN2 <<
VirusPopulationN0;

VirusPopulationN1 = (increaseVirusPopulation_N1,2) >>
VirusPopulationN1 + sendVirusfromN0toN1 >>
VirusPopulationN1 + sendVirusfromN1toN3 <<
VirusPopulationN1 +sendAntivirusfromN0toN1 <<
VirusPopulationN1;

VirusPopulationN2 = (increaseVirusPopulation_N2,2) >>
VirusPopulationN2 + sendVirusfromN0toN2 >>
VirusPopulationN2 + sendAntivirusfromN0toN2 <<
VirusPopulationN2;

VirusPopulationN3 = (increaseVirusPopulation_N3,2) >>
VirusPopulationN3 + sendVirusfromN1toN3 >>
VirusPopulationN3 + sendAntivirusfromN1toN3 <<
VirusPopulationN3;
```

Fig.5. Bio-PEPA Eclipse Plug-in syntax for the virus species definition

fMA takes one parameter, r (in our case r = M), with the overall rate for the reaction being the product of the rate and the population counts of all the reactants and modifier species involved in the reaction [7]. The kinetic laws defined in order to model the virus species actions are shown in Fig.6.

One important feature of the immune system is the capability to self-detect and even remove the intruders. Once a virus has been detected, if the membrane does not contain any antibodies, it tries to procure antibodies and sends a request to the parent membrane. The current approach considers that the skin membrane has unlimited resources of antibodies. In this case, the membrane receiving an antibody request and that does have a resource of antibodies decides to provide the requested antibody for the membrane where it is needed.

```
//Parameters
a = 0.05;
c = 0.1;

//Functional rates (kinetic laws)
increaseVirusPopulation_N0 = [fMA(a)];
increaseVirusPopulation_N1 = [fMA(a)];
increaseVirusPopulation_N2 = [fMA(a)];
increaseVirusPopulation_N3 = [fMA(a)];

sendVirusfromN0toN1 = [c*(VirusPopulationN0@N0-1)];

sendVirusfromN0toN2 = [c*(VirusPopulationN0@N0-1)];

sendVirusfromN1toN3 = [c*(VirusPopulationN1@N1-1)];
```

Fig.6. Bio-PEPA Eclipse Plug-in syntax for the virus species kinetic laws definitions

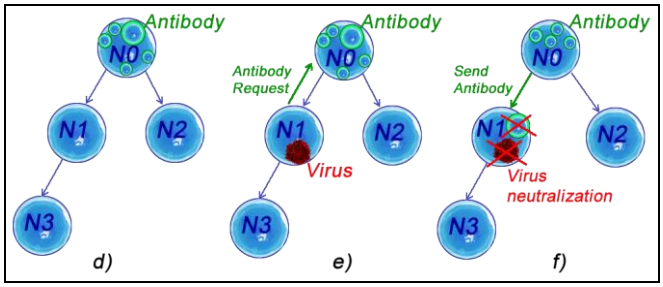

Fig.7. Virus neutralization process

When an antibody meets a virus the neutralization process occurs. This process is presented in Fig.7 and during this reaction both the virus and the antibody are destroyed. The immune system is modeled by defining two antibody species that represent the antibody populations that reside within nodes N0 and N1: *AntiVirusPopulationN0* and *AntiVirusPopulationN1*. In order to keep the system as simple as possible, this approach takes into consideration only those antibody populations within membranes N0 and N1 (because these are the only nodes that contain other nodes).

The definitions of the two antibody species and the associated kinetic laws are presented in Fig.8. For example, the *AntiVirusPopulationN0* population is involved in several reactions: *sendAntivirusfromN0toN1*, *sendAntivirusfromN0toN2*, *increaseAntivirusPopulationN0* and *decrementAntivirusPopulationN0* describing the dynamics of sending an antibody to the interior membranes, auto-increasing the number of antibodies and decrementing the number of antibodies. The neutralization processes that occur within nodes N1, N2 and N3 are managed by the kinetic laws *sendAntivirusfromN0toN1*, *sendAntivirusfromN0toN2* and *sendAntivirusfromN1toN3*. These kinetic laws involve a decrement of the virus species concentration and are followed by the decrement of the antibody species concentration.

Functional rates are introduced into the language in order to express general kinetic laws. Each action type is a reaction and it is associated with a functional rate [7]. Within the proposed Bio-PEPA immune system model the functional rate of the kinetic laws depends on a series of predefined parameters. For example, the *increaseAntivirusPopulationN0* kinetic law is defined based on the mass-action functional rate (fMA) which depends on *b* parameter. Other kinetic laws are defined in a similar way: *increaseAntivirusPopulationN1*, *increaseVirusPopulation_N0*, *increaseVirusPopulation_N1*, *increaseVirusPopulation_N2*, *increaseVirusPopulation_N3*. The virus propagation kinetic laws depend on another parameter, *a*. As the value of this parameter gets higher, the reaction occurs faster. These parameters have an important role in our proposed Bio-PEPA model of the immune system because their values affect the functional rates of the virus propagation and virus neutralization biochemical reactions.

The model component is always the final definition in a Bio-PEPA model [17] and it describes the synchronization process between components.

```
//Parameters
b = 0.02;
d = 0.3;
e = 0.5;

//Functional rates (kinetic laws)
increaseAntivirusPopulationN0 = [fMA(b)];
increaseAntivirusPopulationN1 = [fMA(b)];

sendAntivirusfromN0toN1 = [d*(VirusPopulationN1@N1-1)];

sendAntivirusfromN0toN2 = [d*(VirusPopulationN2@N2-1)];

sendAntivirusfromN1toN3 = [d*(VirusPopulationN3@N3-1)];

decrementAntivirusPopulationN0 =
[e*(AntiVirusPopulationN0@N0-1)];

decrementAntivirusPopulationN1 =
[e*(AntiVirusPopulationN1@N1-1)];

//Species definition
AntiVirusPopulationN0 =   sendAntivirusfromN0toN1 >>
AntiVirusPopulationN0 + sendAntivirusfromN0toN2 >>
AntiVirusPopulationN0 + (increaseAntivirusPopulationN0,2) >>
AntiVirusPopulationN0 + (decrementAntivirusPopulationN0,2)
<< AntiVirusPopulationN0;

AntiVirusPopulationN1 =
sendAntivirusfromN1toN3 >> AntiVirusPopulationN1 +
(increaseAntivirusPopulationN1,2) >>
AntiVirusPopulationN1 + (decrementAntivirusPopulationN1,2)
<< AntiVirusPopulationN1;
```

Fig.8. Bio-PEPA Eclipse Plug-in syntax for the antibody species and associated kinetic laws definitions

The Bio-PEPA implementation of the proposed model component is described in Fig.9. In the current case it comprises three sub-models. The term $P < L > Q$ denotes the cooperation between P and Q over the cooperation set L and determines those activities during which the species are forced to synchronize. This process is called the cooperation process of two species and it is mentioned both in the definition of the involved species and in the component model. For example, if a virus spreads from the root membrane to N1, then a cooperation process is defined between *VirusPopulationN0* and *VirusPopulationN1* species over the kinetic law *sendVirusfromN0toN1*. In this case, the cooperation process is expressed by the model sub-component "*VirusPopulationN0<sendVirusfromN0toN1>VirusPopulationN1*". The kinetic law *sendVirusfromN0toN1* is included in both *VirusPopulationN0* and *VirusPopulationN1* species declarations.

```
// Model component
PopulationN0ToN1 ::=
VirusPopulationN0@N0[1]<sendVirusfromN0toN1>VirusPopulationN1@N1[0];

PopulationN0ToN2 ::=
AntiVirusPopulationN0@N0[1]<sendAntivirusfromN0toN2>VirusPopulationN2@N2[0];

PopulationN1ToN3 ::=
VirusPopulationN3@N3[0]<sendAntivirusfromN1toN3>AntiVirusPopulationN1@N1[0];

PopulationN0ToN2<sendAntivirusfromN0toN1>PopulationN0ToN1<sendVirusfromN1toN3>PopulationN1ToN3
```

Fig.9. Bio-PEPA Eclipse Plug-in syntax for the immune system model component

Within this reaction the role of the species is easy to be determined: one item of the *VirusPopulationN0* species is the *reactant* and changes his location from N0 location to N1 membrane, where it becomes a new item of *VirusPopulationN1* species and represents the product of the reaction. The cooperation between the virus species within the skin membrane and N2 membrane and, of course, between the virus species *VirusPopulationN1* and *VirusPopulationN3* can be presented in a similar manner. As already highlighted, *sendAntivirusfromN0toN1* kinetic law assumes that node N1 has already received and auto-detected a virus and sent an antibody request to the root membrane. This kinetic law imposes the cooperation between *VirusPopulationN1* and *AntiVirusPopulationN0*. Receiving the antibody triggers the virus neutralization process of one virus instance within node N1.

The component model consists of several sub-models used for better legibility. The sub-components are joined in the last line of the model presented in Fig.9. The "@" symbol indicates the initial location of the species and it is followed by one of the four location names (N0, N1, N2 or N3). The terms *VirusPopulationN0@N0[1]* and *AntiVirusPopulationN0@N0[1]* specify that at the beginning of the simulation within the skin membrane resides one instance of the *VirusPopulationN0* species and one instance of the *AntiVirusPopulationN0* species. This example assumes that within all the other locations there are no viruses and antibodies and expects their population concentrations to increase as time passes.

IV. Experimental Results

In Section III the membrane computing patterns and the Bio-PEPA process algebra are used in order to model the response of the immune system to virus attacks. The current section analyzes the behavior of the immune system based on the formal model proposed in the previous section. Several time series analyzes are going to be performed in order to identify the formal conditions expressed based on Bio-PEPA functional rates for successful immune responses.

The relative amount of the virus and antibody biochemical species defined in the proposed formal model will change in time because the concentrations of the species involved in the reactions also change. A time series analysis is going to be used in order to illustrate the evolution of the species concentrations. Such a time series analysis can be applied to a Bio-PEPA formal model in order to simulate the model and plot the quantities of the chemical species in the model as a function of time. Different types of time series simulations are available in the Bio-PEPA Eclipse Plug-in, including continuous, deterministic simulators and stochastic simulators. The deterministic approaches convert the Bio-PEPA model into a system of Ordinary Differential Equations (ODE) and evaluate the model using numerical integration. Stochastic simulators convert the Bio-PEPA model into a Monte Carlo Markov Chain (MCMC) problem which is evaluated using exact or approximate stochastic simulation algorithms such as Gillespie's Direct Method and Gillespie's τ-leap algorithm [17]. For systems that involve large molecular counts, ODE models provide an accurate picture of their behaviour. If molecular counts are small (discrete), then the stochastic analysis may have significant influence on the observed behaviour. Genetic circuits typically involve small molecule counts [12]. The results of both stochastic and ODE simulation of the immune system are analyzed below.

Fig.10. presents a stochastic analysis of the Bio-PEPA formal model of the immune system. Accurate analysis sometimes requires a stochastic process description. A Markov process is one where the next state is only dependent on the present state and not the past history. A stochastic model is a jump Markov process in which the state updates take place in discrete amounts [12]. The graphic illustrates the response of the system in the context of some predefined formal conditions expressed by the parameters of the functional rates (kinetic laws). Functional rates are introduced into the language in order to express the general kinetic laws. Each action type is a reaction and it is associated with a functional rate [7]. Fig.10. presents the possible evolution of the viruses and antibodies concentration of species based on several predefined parameter values. The specified parameters influence the functional rate of the biochemical reactions that occur in the immune system (e.g. virus replication and propagation, antibody population incrementation, virus neutralization). For example, these parameters can be seen as predefined data that describe the initial state of the immune system (e.g. features of the human body's immune system). If the latency of the antibody replication process and the type of the virus (which indicates the maturity and the propagation period of the virus) are known, then the values of the parameters can be easily set. In the following examples we use different values of the predefined parameters. Based on the graphics illustrated in Fig.10 and Fig.11 we can identify the formal conditions that determine an unfavorable immune response. We associate the unfavorable immune response with the system state called "infected". The parameters "a" and "b" influence those kinetic laws that model the behavior of the concentration of viruses within membranes, while the other parameters affect the

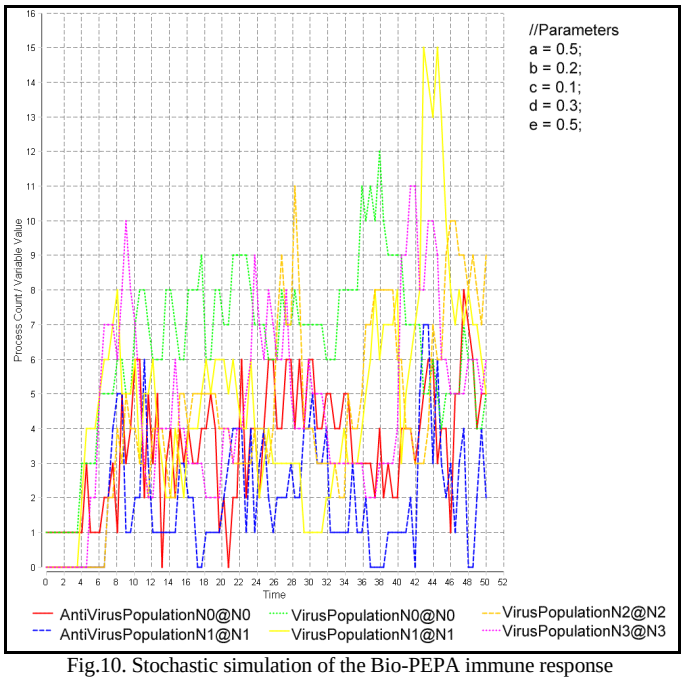

Fig.10. Stochastic simulation of the Bio-PEPA immune response

functional rate of the biochemical reactions that involve the populations of antibodies. If the kinetic law that models the dynamics of the virus propagation process uses a parameter with greater value than the one that models the virus neutralization process, then it is obvious that the propagation reaction will execute faster than the neutralization one. Therefore, parameter "a" influences the virus replication reaction and parameter "c" affects the virus propagation biochemical reaction, as presented in Fig.6. Parameter "b" influences the rate of the antibody replication reaction, while the other two parameters ("d" and "e") affect the virus neutralization process, respectively the reactions that imply the decrement of the antibody population.

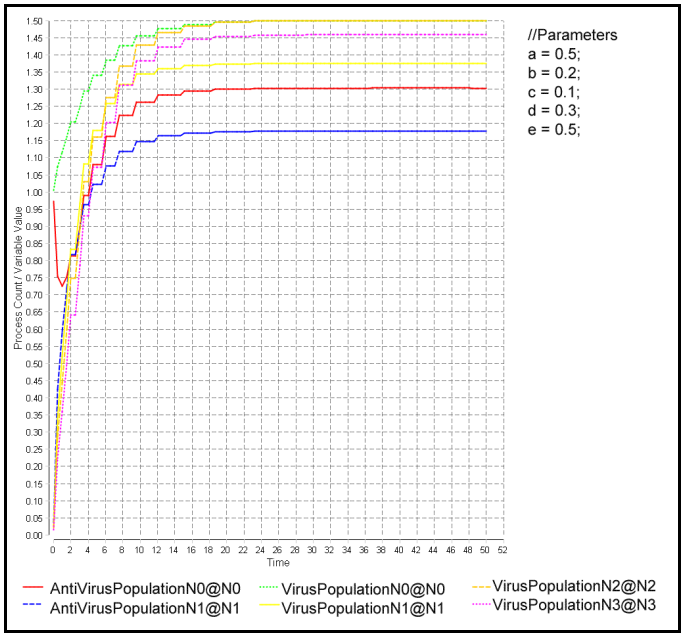

Fig.11. ODE simulation of the Bio-PEPA immune response model

Since the kinetic law describing the virus replication process depends on parameter “a“ and “a” has a greater value than parameter “d”, which influences the neutralization process, we can assume that in time, the concentration of the virus population within the membranes will be greater than the concentration of the antibody population. Fig.11 illustrates the results of the ODE simulation of the Bio-PEPA immune system. ODE analysis is sometimes preferred compared to the stochastic one because it illustrates the evolution of the species using smoother curves. Both Fig.10 and Fig.11 predict a system state that includes high values of the concentration of the virus species. In other words, the system reaches the “infected” final state. In pharmaceutical terms, these two graphics indicate that the body requires external aid provided through medication. The next two figures, Fig.12 and Fig.13, present the results of the stochastic and ODE simulations, but with different values of the predefined parameters. The last two figures show the formal conditions that guarantee the success of the immune system against virus attacks.

Determining whether the virus will be able to infect most of the cells is not a trivial problem. In this example it looks like in the first period of time the antibody population AntiVirusPopulationN0 has greater values of the species concentration compared to the concentration of the virus populations. This indicates a high probability that the system will reach the “clean” state. Interpreting the graphic in Fig.13 one can say that the probability of the antibodies to overcome the viruses is quite high as the virus population within the skin membrane tends to be mitigated as time passes.

The sufficient condition for the system state to be “clean” can be written as M > 2D, where M represents the maturity period of the virus and D is the maximum distance between a virus and the skin membrane [4]. It is easy to see that it will take exactly 2D time for a membrane to receive the antibody after it has requested it. If the above condition is satisfied, then the virus will be destroyed before it reproduces. The parameters used for the simulation of the successful response of the immune system (Fig.12. and Fig.13.) take into consideration this condition: the value of the parameter “a” which affects the virus replication process is lower than the values of the parameters “d” and “b” which influence the virus neutralization and antibody generation processes. The values of these parameters are inverse proportional with M and P. This condition has to be true for each membrane in the system in order for the system to eliminate the viruses. From a pharmaceutical point of view, this means that the host organism (e.g. the human body) is strong enough to remove the intruders without any external help.

However, the necessary condition is a lot more complicated [4]. While the number of viruses in the system is increasing as time passes, generalizing we obtain the equation (2), where t represents the time, P is the virus propagation period and M is the virus maturity period. By performing a series of complex analysis like the time series analysis presented in this section, we can predict the evolution of the system, the possible inconsistencies or we can even validate the proposed formal model.

$$n(t) = n(t-P) + n(t-M) \quad (2)$$

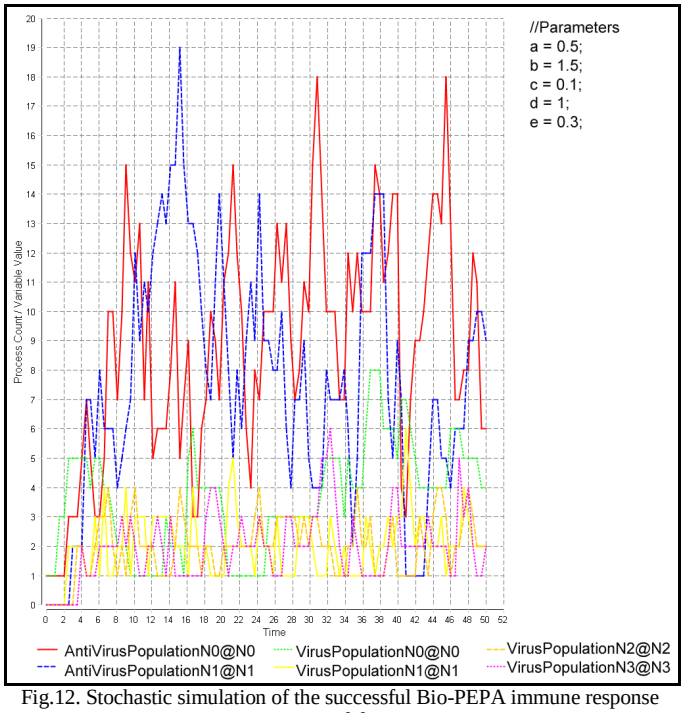

Fig.12. Stochastic simulation of the successful Bio-PEPA immune response model

As shown in the previous stochastic and ODE simulations results, the parameters involved in the biochemical reactions used to model the immune response are very important in the system. These parameters affect the execution time of all reactions. The execution time of the reactions is very important in determining the possible final state of the system (“clean” or “infected”).

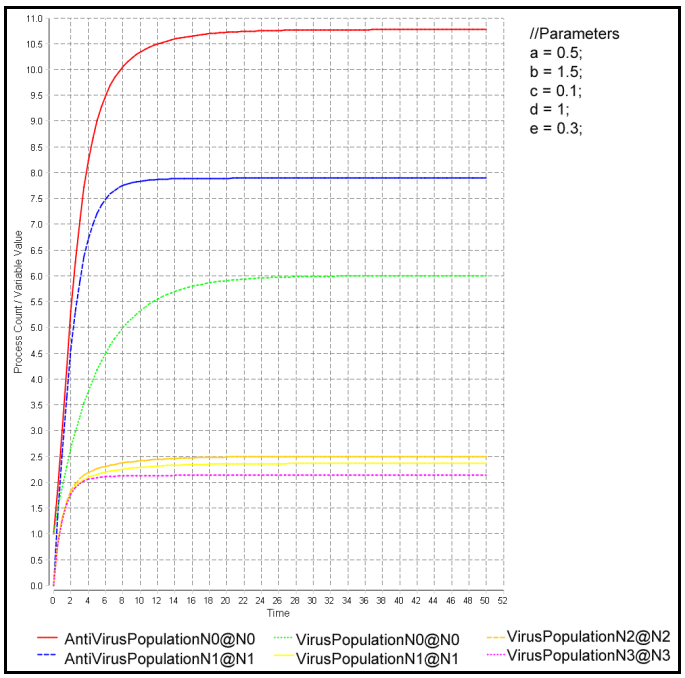

Fig.13. ODE simulation of the successful Bio-PEPA immune response model

From a pharmaceutical point of view, this means that it is necessary to know the type of the virus and the level of the immune system in order to set the values of these parameters. If the immune system has a low level of immunity, then the virus detection and neutralization reactions will be executed in a longer period of time. Similar, if the immune system has a high level of immunity, then the execution time of the virus neutralization process is shorter. These information are likely to be used as parameters in the simulation of the Bio-PEPA immune response formal model developed in Section III.

V. FUTURE WORK

In the near we intend to continue our work with a Java parallel implementation of the immune system model developed and analyzed in this paper. This system will map each membrane on a core of a multicore computer and will try to simulate the inter-cellular communication and synchronization, as well as the virus and antibodies interaction, replication and neutralization processes. The purpose of the Java implementation is to validate the stochastic and ODE simulations achieved using Bio-PEPA.

We also intend to investigate the semantics of Bio-PEPA by using methods in the tradition of programming language semantics, namely operational and denotational semantics. We will use continuations semantics for concurrency [15,6], which can describe distributed systems with dynamic configurations [5,16], specific of membrane computing and other biologically inspired models of computation.

VI. CONCLUSIONS

The paper proposes a strategy of modeling and analyzing the behavior of the immune system based on membrane computing patterns and the Bio-PEPA stochastic process algebra. In terms of distributed computing we can say that the immune system is an environment with a high degree of concurrency. In this environment it is possible that the amount of the existing resources to be deficient compared to the demand for resources and that's why a continuous activity to protect the system is mandatory. The Bio-PEPA experimental results given in this paper show that the effectiveness of modeling biochemical networks can be significantly improved by providing an intermediate, formal compositional representation of the model on which different kinds of analysis can be carried out.

REFERENCES

[1] A. Alexandru, G. Ciobanu., "Mathematics of multisets in the Fraenkel-Mostowski framework," Bull. Math. Soc. Sci. Math. Roumanie, tome 58(106), no.1, 3–18, 2015.

[2] B. Aman, G. Ciobanu, "Mobility in Process Calculi and Natural Computing," Springer, 2011.

[3] G. Caravagna, J. Hillston, "Modeling biological systems with delays in BioPEPA," Proc. of the 4-th Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC 2010), EPTCS 40 pp. 85–101, 2010.

[4] G. Ciobanu, "Membrane Computing & Biologically Inspired Process Calculi," Editura Universitatii "Alexandru Ioan Cuza" Iasi, 2010

[5] G. Ciobanu, E.N. Todoran, "Relating two metric semantics for parallel rewriting of mulisets," Proceedings of 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012), pp. 273-280, IEEE Computer Press, 2012.

[6] G. Ciobanu, E.N. Todoran, "Continuation semantics for asynchronous concurrency," Fundamenta Informaticae, vol. 131(3-4), pp. 373–388, 2014.

[7] F. Ciocchetta, J. Hillston, "Bio-PEPA: a Framework for the Modelling and Analysis of Biological Systems," Theoretical Computer Science, 410(33-34):3065-3084, 2009

[8] A. Clark, S. Gilmore, M. L. Guerriero, J. Hillston, "Conservation of Mass Analysis for Bio-PEPA," Electronic Notes in Theoretical Computer Science, 2012.

[9] A. Duguid, "An overview of the Bio-PEPA Eclipse plug-in," PASTA 2009, 8th Workshop on Process Algebra and Stochastically Timed Activities, pages 121-131, 2009.

[10] J. Hillston, "A Compositional Approach to Performance Modelling," Cambridge University Press, 1996

[11] D. Marco, E. Scott, D. Cairns, A. Graham, J. Allen, S. Mahajan, C. Shankland, "Investigating co-infection dynamics through evolution of Bio-PEPA model parameters: A combined process algebra and evolutionary computing approach," in D. Gilbert, M. Heiner (eds.) CMSB 2012. LNCS, vol. 7605, pp. 227–246. Springer, Heidelberg, 2012

[12] C. J. Myers, "Modeling and Analysis of Biological Networks," ECE/CS/BioEn 6760, http://www.async.ece.utah.edu/~myers/nobackup/ece6760_07/lectures/lec9-2x3.pdf

[13] Gh. Păun, "Introduction to Membrane Computing," in G. Ciobanu, Gh. Păun, M. J. Pérez-Jiménez (eds.) Applications of Membrane Computing, pages 1-46, Springer, 2006.

[14] Gh. Păun, Membrane Computing. An Introduction. Springer, 2002.

[15] E.N. Todoran, "Comparative semantics for modern communication abstractions," Proceedings of 2008 IEEE 4th International Conference on Intelligent Computer Communication and Processing (ICCP 2008), pp. 153-161, 2008.

[16] E.N. Todoran, N. Papaspyrou, "Experiments with continuation semantics for DNA computing," Proceedings of 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP 2013), pp. 251-259, 2013.

[17] The Bio-PEPA Eclipse Plug-In User Manual, http://www.biopepa.org, 2012

[18] Bio-PEPA modeling language, http://www.biopepa.org