System Dynamics Metamodels Supporting the Development of Computational Models of the Human Innate Immune System

Igor Knop[1], Alexandre Pigozzo[1],
Barbara Quintela[1], Gilson C. Macedo[2],
Ciro Barbosa[1], Rodrigo Weber dos Santos[1], and Marcelo Lobosco[1]

[1] FISIOCOMP, Universidade Federal de Juiz de Fora
[2] Immunology Lab, Universidade Federal de Juiz de Fora

Abstract. The human body is protected against pathogenic invasions by a complex system of cells, tissues and organs which form the Human Immune System (HIS). Understanding how the HIS works is therefore essential to obtain new insights into its nature and to deal effectively with diseases. Mathematical and computational modeling can be used for this purpose. Unfortunately, these complex mathematical models are very difficult to develop, understand and use by a more general and multidisciplinary team. This paper presents a System Dynamics Metamodeling tool, called JynaCore API, that supports the development of complex models using System Dynamics in a more abstract level. To demonstrate the power and usefulness of the proposed System Dynamics Metamodeling tool, in this work we present the development of a complex two-dimensional tissue model that simulates the dynamics of the immune response.

1 Introduction

The Human Immune System (HIS) is composed of two distinct parts, the innate immune system and the adaptive immune system[1]. The innate immune system is responsible for powerful nonspecific defenses that prevents or limits infections by most pathogenic microorganisms. This first line of defense against pathogenic microorganisms consists of physical barriers, such as the skin and mucous membranes, and the second line consists of cells, such as neutrophils, that recognize specific parts of pathogenic microorganisms, herein called antigens. Understanding how the HIS works is essential to obtain new insights into its nature and to deal effectively with diseases. However, the multiscale and multiphysics nature of the involved phenomena poses great challenges to the researchers of this field. Computational modeling can be used for perform experiments in silico, allowing researchers to speed up the discovery of new, cheaper and more effective drugs against diseases. In addition, computer models have become valuable tools for the study and comprehension of such complex phenomena, as they allow different information acquired from different physical scales and experiments to be combined to generate a better picture of the whole system functionality. Not surprisingly, the high complexity of the phenomena translates into complex mathematical and computational models. Computational models of the HIS are very often developed using pure mathematical tools, such as nonlinear systems of partial differential equations, to describe the behavior of its components and their relationships. Unfortunately, these complex mathematical models are very difficult to develop, understand and use by a more general and multidisciplinary public. The complexity of the models has limited the number of research centers that develop and make efficient use of them.

In this paper we propose a solution for some of these challenges: JynaCore API, a System Dynamics Metamodeling tool that supports the development of complex models using System Dynamics in a more abstract level. Two levels of abstraction are used: domain modeling and instance modeling. First, instead of using pure mathematical equations, i.e. system of nonlinear ordinary equations, System Dynamics is used to describe the relations between the different variables of the model. System Dynamics uses stock, flux diagrams and simple equations to describe systems with complex dynamic behavior. A stock and flux diagram are a formal and quantitative way to express the main concepts of a system and their structural relations. Second, after a System Dynamics model is created, i.e. a domain model, one can instantiate it multiple times and connect the different instances with relations that are freely defined. To demonstrate the power and usefulness of our System Dynamics Metamodeling tool, in this work we present the development of a complex two-dimensional tissue model that simulates the dynamics of the immune response and that include important features such as diffusion and chemotaxis behavior of neutrophils.

2 System Dynamics

The System Dynamics use stock and flux diagrams and simple equations to describe systems with complex dynamic behavior. A stock and flux diagram are a formal and quantitative way to express the main concepts of a system and their structural relations. In System Dynamics, a stock and flux diagram is composed by four basic elements: stock; flux; auxiliaries and information. A stock express a quantity that can be accumulated in our system. In our model, for instance, we can express populations of neutrophils as stocks. A flux express how stock level varies in time. A stock level can only change due to a incoming flux (raising up an amount stored) or outgoing flux (lowering down an amount stored). An auxiliary is a way to isolate important data from the system. An auxiliary cannot directly change a stock level, but can be used to calculate a flux value. Information comes as visual representation in the model. When a flux or auxiliary uses a stock level in their equations, we trace a thin arrow from stock to flux or variable. With that visual aid we can see the feedback loops which increases the system understanding.

Fig. 1 presents an example of the basic stock-flux connection. The cloud symbol is an Infinite Stock that represents an accumulation that is out of our interest in that specific model. Infinite stocks are considered capable of giving or receiving any value from fluxes. Hidden behind this intuitive and graphical description is a mathematical model based on an ODE.

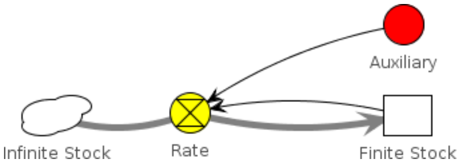


Fig. 1: System Dynamics basic elements.

3 System Dynamics Metamodels

A model should answer a set of questions in a language which the user understands. System Dynamics is a formal language that can be used in multidisciplinary group as knowledge exchanging tool. All basic elements and relations of sub systems can be captured and associated together to system behaviour that emerges in a endogenous way. However the readability diminishes when more and more elements from sub systems are added. Marcio Barros proposed the System Dynamics Metamodeling[2], a new modeling language based on System Dynamics to limit the growing complexity, leverage readability and maintainability by dividing the model in two layers of interest.

The System Dynamics Metamodeling split model construction in two phases: domain modeling and instance modeling. A domain modeling describes behavior structures through classes. The Fig.2 shows the relation between a domain model and an instance model. An instance model uses previously defined classes to create instance elements which can be associated and have properties changed to describe a specific system. An optional step in domain modeling consists of creating scenario models to encapsulate some structural model changes. Those scenarios can be used in various instance models to as an repeatable and encapsulated experiment.

3.1 Domain Model

A domain model try to capture properties, behavior and relation between a set of elements of a knowledge domain. It is an abstract description, therefore, it can not be realized in an simulation. To construct a domain model, the modeler should know the stock and flux diagrams from System Dynamics.

The main concept from domain models is the *class*. A class is a representation of a family of elements who owns a same behavior and properties. From a class we can describe instances, each one with its own states. To describe a class we use a stock and flux diagram as a complete and isolated system. We can

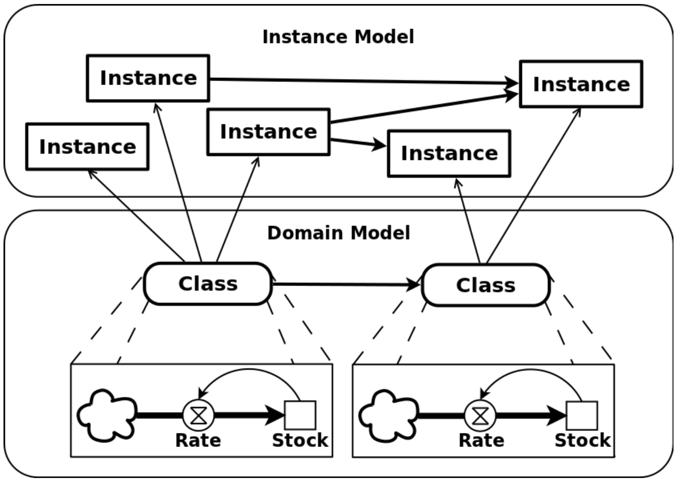
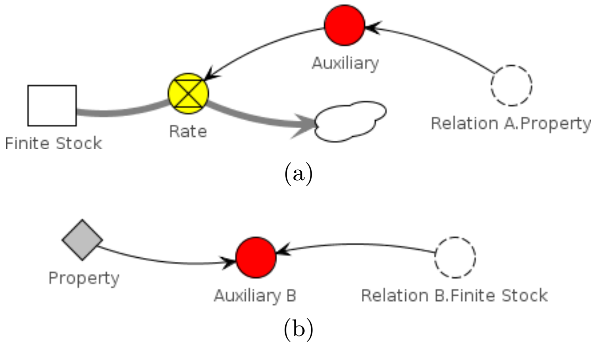


Fig. 2: System Dynamics Metamodeling. A domain model define behavior structures in classes and a instance model define individuals that implement those classes and their interrelations to model a specific system.

define relations between classes that allow us to access external elements defined elsewhere.

For instance, as a simple example, we can consider to create two classes: "Class A" and "Class B". Class A in Fig.3(a) has a finite stock, a rate and an auxiliary. Class B in Fig.3(b) has only an auxiliary and a property. The dashed auxiliaries are external elements accessed through relationships.



(a)

(b)

Fig. 3: System Dynamics Domain Model for two classes.

3.2 Instance Model

Only after defining a domain model we can start modeling our system using domain classes. An instance model is a realization of the domain model on which we construct elements based on the available class structure.

In using our domain model, we can consider to create simple instance model by instancing our domain classes and connecting them properly. Fig.4 shows a simple example of an instance model.

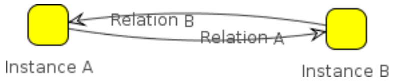


Fig. 4: System Dynamics basic elements.

4 HIS

Body surfaces are protected by epithelia, which constitutes a physical barrier between the internal and external environments. The body's epithelia form an effective block against the external environment, but eventually they can be crossed or settled by pathogens, causing infections. After crossing the epithelium, the pathogens encounter cells and molecules of the innate immune system that immediately develop a response. Reinforcing the innate immune response but taking days instead of hours to develop, the adaptive immune system is capable of eliminating the infection more efficiently than the innate immune system. The adaptive immune system is only present in vertebrates and depends primarily on the recognition executed by lymphocytes, that posses the ability to distinguish a pathogen and direct to it a stronger immune response.

The initial response of the body to an acute biological stress, such as a bacterial infection, is an acute inflammatory response [3]. The strategy of the HIS is to keep some resident cells on guard in the tissues to look for any signal of infection. When they find such a signal, the resident cells of the innate system start to produce substances such that the cytokine that recruit cells and molecules of innate immunity from blood vessels to the location of the infected tissue.

The LPS endotoxin is a potent immunostimulant that can induce an acute inflammatory response comparable to that of a bacterial infection. After the lyse of the bacteria by the action of cells of the HIS, the LPS can be released in the host, intensifying the inflammatory response and activating some cells of the innate system. The LPS can trigger an inflammatory response through the interaction with receptors on the surface of some cells. The commitment of this receptor activates these cells to phagocytose, degrading the bacteria internally and secreting proteins known as cytokines and chemokines, as well as other molecules.

The inflammation of an infectious tissue has many benefits in the control of the infection. Besides recruiting cells and molecules of innate immunity from blood vessels to the location of the infected tissue, it increases the lymph flux containing microorganisms and cells that carry LPS to the neighbors lymphoid tissues, where these cells will present the LPS to the lymphocytes and will initiate the adaptive response. Once the adaptive response is activated, the inflammation also recruits the effector cells of the adaptive immune system to the location of infection.

5 Methods

5.1 Mathematical Model

The model proposed in this work is a system of Partial Differential Equations (PDEs) based on the originally proposed by [4]. In this model, a set of equations describe the dynamics of the immune response to LPS in a microscopic section of tissue. In particular, the interactions among LPS, neutrophil and cytokine are modeled.

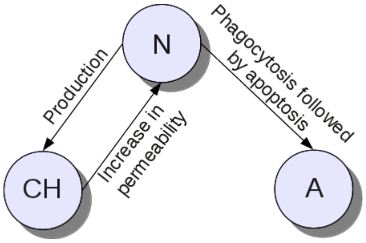


Fig. 5: Relations between the model components.

Figure 5 presents schematically the relationship between neutrophils, pro-inflammatory cytokines and LPS. The LPS diffuse and cause a response in the neutrophils, that recognize these LPS and phagocyte them. The process of phagocytosis induces, in a rapid way, the apoptosis of neutrophils.

The pro-inflammatory cytokine is produced by neutrophils after the membrane receptors of these neutrophils recognize the LPS. The pro-inflammatory cytokine induces an increase in the endothelial permeability allowing more neutrophils to leave the blood vessels and enter the infected tissue. Besides, the pro-inflammatory cytokine is chemoattractant of neutrophils, guiding their movement. As a result the neutrophils move in the direction of the gradient of the pro-inflammatory cytokine.

The main characteristics of the mathematical model are:

- Neutrophils interact with pro-inflammatory cytokines and LPS;

– The interaction between neutrophils and LPS increases the production of cytokines;
– Cytokines induce an increase in the endothelial permeability and allows more neutrophils to come to the infected tissue;
– In the tissue, neutrophils move in the direction of the gradient of the pro-inflammatory cytokines (chemotaxis);
– Pro-inflammatory cytokines attracts the neutrophils to the location where the LPS concentration is higher.

Our set of equations is given below, where A, N and CH represent the population of LPS, neutrophils and pro-inflammatory cytokines, respectively.

The LPS equation is shown in Equation 1.

$$\begin{cases} \frac{\partial A}{\partial t} = -\mu_A A - \lambda_{N|A} A.N + D_A \Delta A \\ A(x,0) = A_0 \quad | \quad 0 \le x < 1, \frac{\partial A(.,t)}{\partial n}|_{\partial\Omega} = 0 \end{cases} \tag{1}$$

The term $\mu_A A$ models the decay of the LPS, where μ_A is its decay rate. The term $\lambda_{N|A} A.N$ models the phagocytosis of LPS by neutrophils, where $\lambda_{N|A}$ is the phagocytosis rate. The term $D_A \Delta A$ models the diffusion of the LPS, where D_A is the diffusion coefficient. the amount of substance diffusing across a unit area through a unit concentration gradient in unit time.

The neutrophil equation is shown in Equation 2.

$$\begin{cases} permeability = ((Pmax - Pmin).CH/(CH + Keqch) + Pmin) \\ sourceN = permeability.(NmaxTissue - N) \\ \frac{\partial N}{\partial t} = -\mu_N N - \lambda_{A|N} A.N + D_N \Delta N + sourceN - \nabla.(\chi_N N \nabla CH) \\ N(x,0) = N_0, \frac{\partial N(.,t)}{\partial n}|_{\partial\Omega} = 0 \end{cases} \tag{2}$$

The term $((Pmax - Pmin).CH/(CH + Keqch) + Pmin)$ uses a Hill equation [5] to model how permeability of the endothelium of the blood vessels depends on the local concentration of cytokines. Hill equations are also used, for example, to model drug dose-response relationships [6]. The idea is to model the increase in the permeability of the endothelium according to the concentration of the pro-inflammatory cytokines into the endothelium. In the Hill equation, $Pmax$ represents the maximum rate of increase of endothelium permeability induced by pro-inflammatory cytokines, $Pmin$ represents the minimum rate of increase of endothelium permeability induced by pro-inflammatory cytokines and $keqch$ is the concentration of the pro-inflammatory cytokine that exerts 50% of the maximum effect in the increase of the permeability.

The term $\mu_N N$ models the neutrophil apoptosis, where μ_N is the rate of apoptosis. The term $\lambda_{A|N} A.N$ models the neutrophil apoptosis induced by the phagocytosis, where $\lambda_{A|N}$ represent the rate of this induced apoptosis. The term $D_N \Delta N$ models the neutrophil diffusion, where D_N is the coefficient of diffusion. The term $sourceN$ represents the source term of neutrophil, that is, the number of neutrophils that is entering the tissue from the blood vessels. This number depends on the endothelium permeability (*permeability*) and the capacity of the tissue to support the entrance of neutrophils ($NmaxTissue$), that can also represent the blood concentration of Neutrophils. In this model we consider it, $NmaxTissue$, constant over time. The term $\nabla.(\chi_N N \nabla CH)$ models the chemotaxis process of the neutrophils, where χ_N is the chemotaxis rate.

Finally, the cytokine equation is shown in Equation 3.

$$\begin{cases} \frac{\partial CH}{\partial t} = -\mu_{CH} CH + \beta_{CH|N}.N.A + D_{CH} \Delta CH \\ CH(x,0) = 0, \frac{\partial CH(.,t)}{\partial n}|_{\partial\Omega} = 0 \end{cases} \tag{3}$$

The term $\mu_{CH} CH$ models the pro-inflammatory cytokine decay, where μ_{CH} is the decay rate. The term $\beta_{CH|N}.N.A$ models the production of the pro-inflammatory cytokine by the neutrophils, where $\beta_{CH|N}$ is the rate of this production. The term $D_{CH} \Delta CH$ models the diffusion of the pro-inflammatory cytokines, where D_{CH} is the diffusion coefficient.

Table 1 presents the initial conditions and the values of the parameters used in the simulations.

Table 1: Initial conditions and parameters

Parameter	**Value**	**Unit**	Reference
N_0	$2, 0 < x < 5$	cell	estimated
CH_0	$0, 0 < x < 5$	cell	estimated
A_0	$50, 0 < x < 1$	cell	estimated
$Pmax$	10	1/day	estimated based on [7]
$Pmin$	1	1/day	estimated based on [7]
$NmaxTissue$	5	cell	estimated
$keqch$	1	cell	estimated
μ_A	0.005	1/day	[8]
$\lambda_{N\|A}$	0.55	1/cell.day	[8]
D_A	2000	μm^2/day	estimated
μ_N	3.43	1/day	[8]
$\lambda_{A\|N}$	0.55	1/cell.day	[8]
X_N	14400	μm^2/day	[9]
D_N	12096	μm^2/day	[10]
μ_{CH}	7	1/day	estimated
$\beta_{CH\|N}$	0.4	1/cell.day	[8]
D_{CH}	9216	μm^2/day	[8]

5.2 Jynacore API

Any computational modeling research group accumulates a growing number of new software tools and models. Those applications varies from new efficient simulators, proof of concept models and tools. In that highly experimental environment there is a need for flexible software tools that can fit in many use cases not previously foreseen.

There is a high number of commercial software tools to simulate stock and flux models that can be used to create and improve the understanding of a system. However, after model validation its hard to integrate them in new experimental applications due to the lack of an API or restrictive software licenses fees. The Jynacore API is as a open source software licensed in Lesser General Public License (LGPL) that do not provide a full modeling environment but encapsulate all simulation process including model description, use of multiple numerical methods and simulation data filtering. It was designed to allow the creation of new experimental computational modeling software with minimal restrictions. The API's source code comes with a complete Java implementation and two numeric methods to simulate stock and flux models of System Dynamics.

There are two ways to build a new model in the current Java implementation of Jynacore API: a) a simple XML, human readable file with stock and flux diagram elements describing the structure and Content MathML for the equations; b) direct API manipulation via Java classes. In both methods the model is kept in memory for future manipulation and simulations, which allows the user to modify the model during run time. The Jynacore API and its default implementation also allows the use of System Dynamics Metamodeling which split the modeling process in a two domain level: a domain model level, where the stock and flux constructors are used to describe a set of classes; and an instance model level, where a specific system is modeled using the classes previously defined in domain model.

5.3 Jynacore SIM

The JynaSIM is a desktop simulation environment built-in upon the Jynacore API and Java. JynaSIM can simulate stock and flux models and instance models in the case of System Dynamics Metamodeling.

The basic usage of JynaSIM is simple: after the XML file that describes a model is loaded, the user set up the simulation parameters and the numeric method to be used. Then, different scenarios can be simulated. JynaSIM also allows to filter the resulting data and to plot the model behavior on a chart. In addition, JynaSIM automatically extracts the model structured as described in the XML file and draws the diagrams and connections that graphically describe the model, using the System Dynamics language, i.e. stocks, flux, auxiliary variables and information arrows and connections. Figure 6 presents a snapshot of the JynaSIM desktop simulation environment.

6 Results

Our innate HIS model was built using the System Dynamics metamodeling tool.

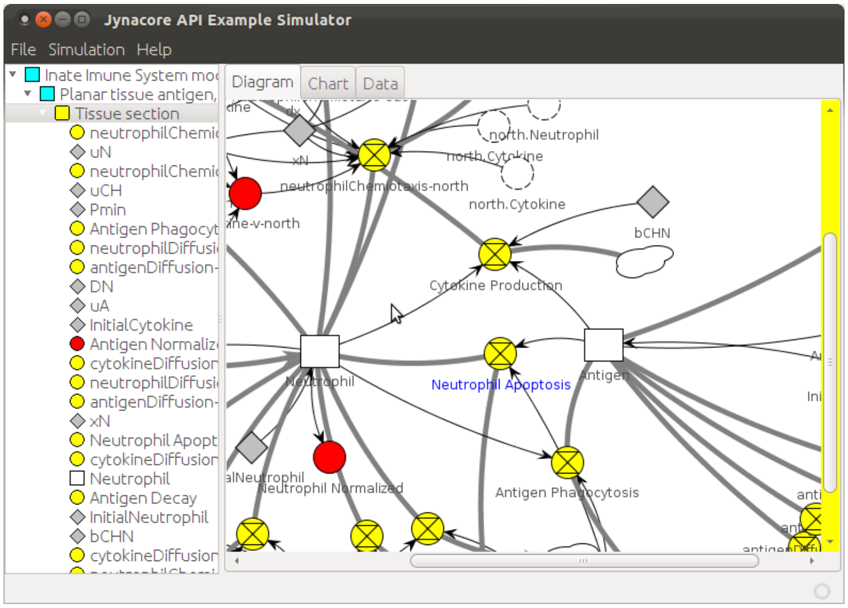


Fig. 6: JynaSIM desktop simulation environment.

6.1 Modeling the antigen, neutrophil and cytokine relations

The dimensionless model presented in Section 5.1 for the dynamics of neutrophils, antigens and cytokines was implemented using the System Dynamics stock and flux diagrams, as shown in Fig. 7.

The Fig.8 shows the dynamics of our model of innate human system with the cytokine production during antigen phagocitosis from neutrophils. We can see the increase of neutrophils due to tissue increased permeability caused by the cytokine. It also shows the decrease of antigens until their complete elimination due to the immune response given by the neutrophils. With the elimination of the antigen the number of neutrophils goes to zero.

In the next sections we will include the spatial features of the model, i.e. diffusion and chemotaxis opearators.

6.2 Modeling diffusion in 1D

In our immune system model, we create a class "Cell" which describes the behavior of a microscopic section of tissue. In order to model an one-dimensional tissue, we use relationships between Cell instances in an discretized space by using "west" and "east" connections between neighboring Cells. For instance, Figure 9 presents how 1D diffusion of antigens is implemented using the System Dynamics Metamodeling tool. Each class "Cell" has two infinite stocks, *east.Antigen* and *west.Antigen* that will be used to calculate the incoming (or

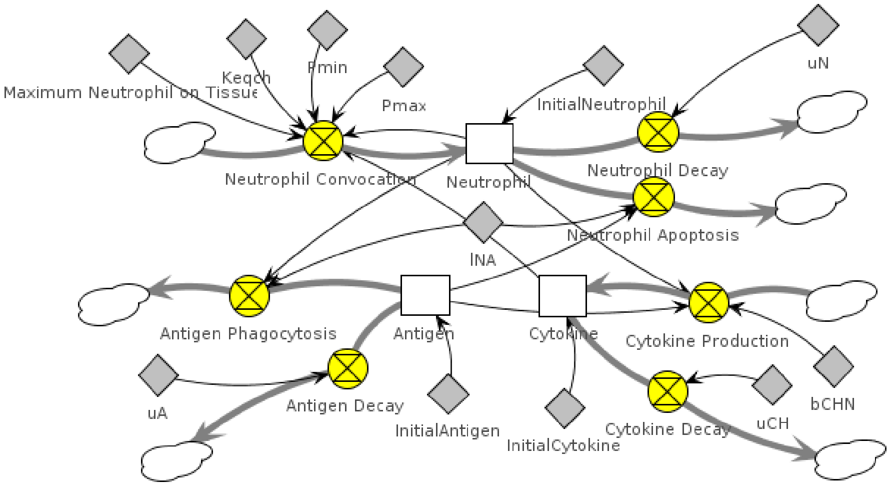


Fig. 7: The System Dynamics model of antigen, neutrophil and cytokine.

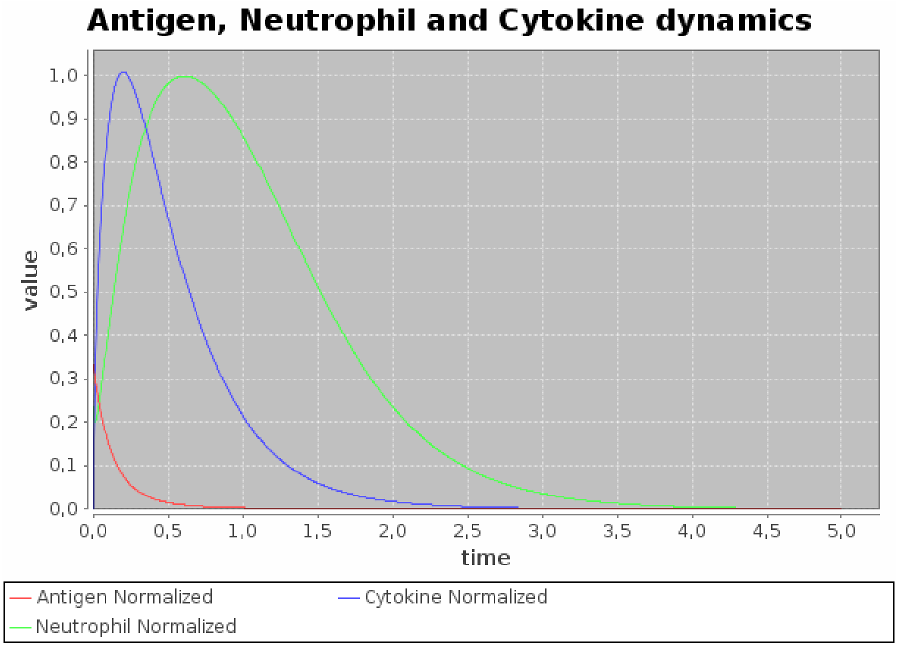


Fig. 8: The dynamics of antigen, neutrophil and cytokine in time.

outgoing) population flux through the boundaries of this tissue volume. The values of the diffusion fluxes depend on the values *Antigen* of the neighbors. To expresses that, we use slashed line auxiliaries that are automatically associated during a relationship.

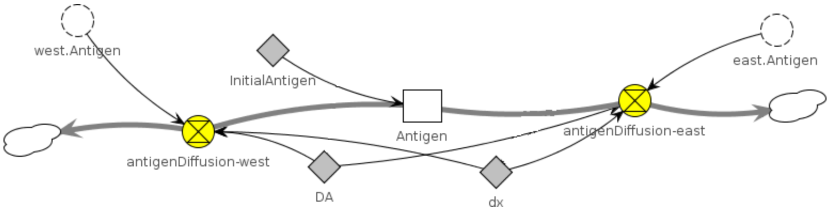


Fig. 9: Diffusion of antigens in Cell volume. We use a stock to model the amount of antigens in this Cell volume and two fluxes to describe the 1D diffusion to neighboring cells.

The initial antigen population is defined by the property *InitialAntigen* which is used as initial value in *Antigen* stock. The diffusion fluxes in Eq.4c and Eq.4d use the properties dx, tissue volume discretization, and D_A, diffusion coefficient of antigens, to calculate the amount of antigen population that leaves or enters the tissue volume. Both fluxes are based on the values of *Antigen* of the neighbors.

$$InitialAntigen = 5.0 \tag{4a}$$

$$Antigen(0) = InitialAntigen \tag{4b}$$

$$antigenDiffusion_{west} = D_A(Antigen - west.Antigen)/dx^2 \tag{4c}$$

$$antigenDiffusion_{east} = D_A(Antigen - east.Antigen)/dx^2 \tag{4d}$$

This method is also used to model neutrophil and cytokine diffusion. This is an intuitive and easy way to explain to newcomers in computational modeling the diffusion process of substances.

The Fig.10 shows a instance model with five Cells disposed in a linear form.

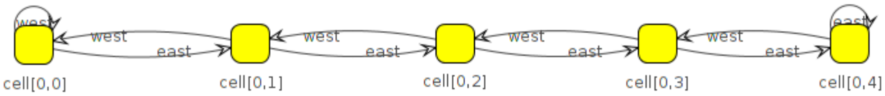


Fig. 10: Instance model with five sections of tissue.

Each one of the class instances can have its own states and properties but the behavior is defined by their internal structure described in the domain class

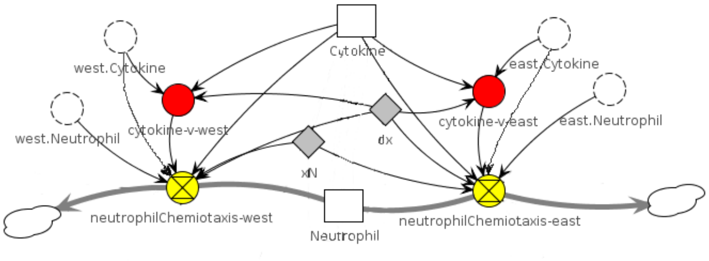


Fig. 11: Domain model of neutrophil chemotaxis for 1D tissue model.

model. If the domain class changes, all instance models based on it have their behaviors changed.

6.3 Modeling the neutrophil chemotaxis in 1D

Similar to the implementation of the diffusion operator, the chemotaxis mathematical operator that models the transport of neutrophils in the direction of the gradient of cytokines can also be implemented using our System Dynamics Metamodeling tool. Each class "Cell" has four infinite stocks, $east.Neutrophil$, $west.Neutrophil$, $east.Cytokine$, $west.Cytokine$ that will be used to calculate the incoming (or outgoing) population flux through the boundaries of this tissue volume. For the chemotaxis, the values of the fluxes depend on the values of the $Neutrophil$ and $Cytokine$ of the neighbors. To expresses that, we use slashed line auxiliaries that are automatically associated during a relationship, see Figure 11.

To calculate the fluxes we also make use of the upwind formulation for finite volume schemes:

$$InitialNeutrophil = 1.0 \tag{5a}$$

$$cytokine_{v-west} = \frac{(west.cytokine - cytokine)}{dx} \tag{5b}$$

$$cytokine_{v-east} = \frac{(east.cytokine - cytokine)}{dx} \tag{5c}$$

$$Neutrophil(0) = InitialNeutrophil \tag{5d}$$

$$neutrophilChemiotaxis_{west} = \begin{cases} \text{if}(cytokine_{v-west} > 0.0) \\ \qquad X_N \cdot \frac{Neutrophil}{dx} \cdot \frac{(west.Cytokine - Cytokine)}{dx} \\ \text{else} \\ \qquad X_N \cdot \frac{west.Neutrophil}{dx} \cdot \frac{(Cytokine - west.Cytokine)}{dx} \end{cases} \tag{6a}$$

$$neutrophilChemiotaxis_{east} = \begin{cases} \text{if}(cytokine_{v-east} > 0.0) \\ \qquad X_N \cdot \frac{Neutrophil}{dx} \cdot \frac{(east.Cytokine - Cytokine)}{dx} \\ \text{else} \\ \qquad X_N \cdot \frac{east.Neutrophil}{dx} \cdot \frac{(Cytokine - east.Cytokine)}{dx} \end{cases} \tag{6b}$$

6.4 Two-dimensional tissue of HIS

We extended the linear model structure to include two other neighbors to every Cell class, or tissue volume: north and south. This allowed us to model and simulate the HIS response on a two-dimensional tissue. In Fig.12 we used the four relationships in the same way as we did in the linear model.

As in the linear model, we can use the same domain class to create a model with any number of sections. Fig.10 shows a model with twenty five sections of tissue distributed in an 5x5 array.

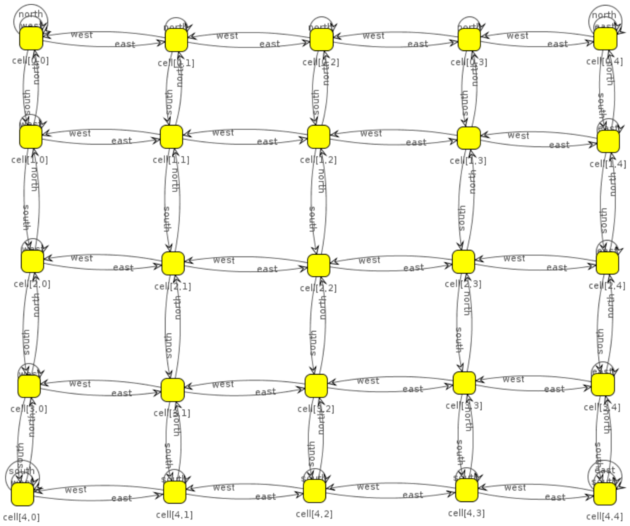


Fig. 12: Instance model with twenty five instances as a two-dimensional tissue.

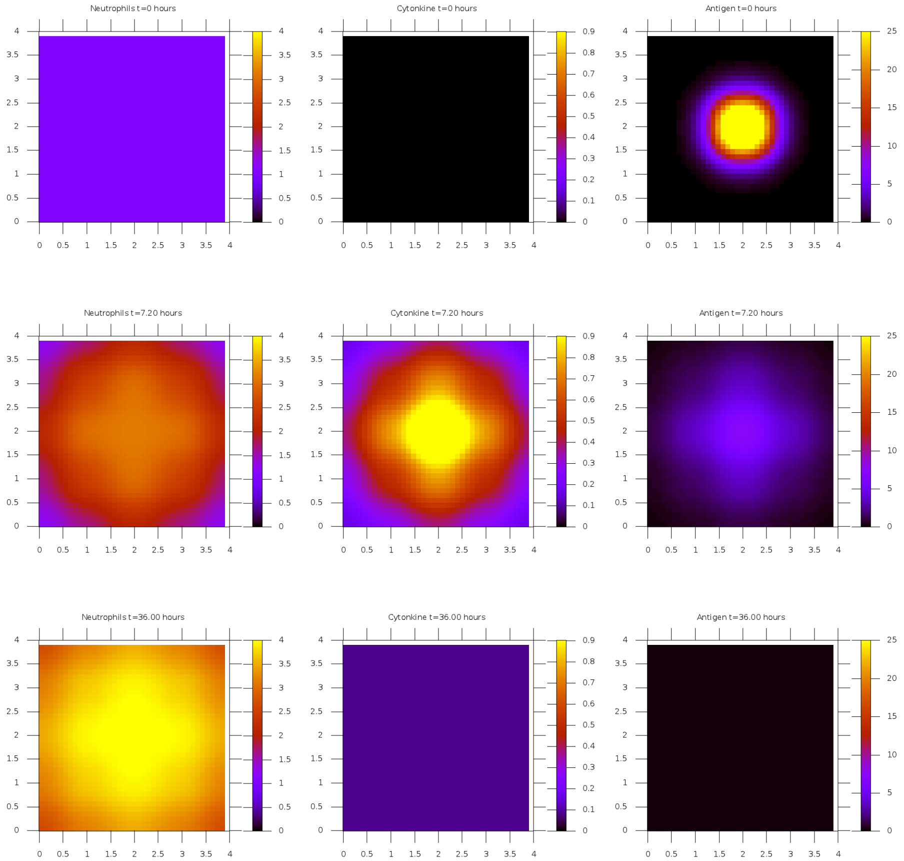


Fig. 13: Neutrophil, cytokine and antigen kinetics on tissue.

Fig. 13 presents neutrophil, cytokine and antigen kinetics on tissue for three different time instants of the simulation. Antigens population decreases due to phagocitosis by neutrophils. The neutrophils release cytokine on the tissue which increases endothelium permeability.

7 Conclusions

In this work we presented a System Dynamics Metamodeling tool, called JynaCore API, that supports the development of complex models using System Dynamics in a more abstract level. Two levels of abstraction are used: domain modeling and instance modeling. First, instead of using pure mathematical equations, i.e. system of nonlinear ordinary equations, System Dynamics is used to describe the relations between the different variables of the model. Second, the

System Dynamics models created can be instantiated multiple times and the relation between each instance can be freely defined. In this way, we could easily develop a two-dimensional tissue model that simulated the dynamics of the immune response and that included important features such as diffusion and chemotaxis behavior of neutrophils. We conclude that this new framework, based on the JynaCore API, can be used for building new models and new in silico experiments to support the development of computational Human Immune Systems.

8 Acknowledgment

The authors would like to thank the Brazilian agencies CAPES, CNPq, FAPEMIG, FINEP and UFJF.

References

1. Paul, W.E.: Fundamental immunology. 6nd edn. Lippincott Williams & Wilkins, Philadelphia (2008)
2. De Oliveira Barros, M., Werner, C.M.L., Travassos, G.H.: A system dynamics metamodel for software process modeling. Software Process: Improvement and Practice **7**(3-4) (2002) 161–172
3. Murphy, K.M., Travers, P., Walport, M.: Imunobiologia. 7 edn. Garland Science (2010)
4. Pigozzo, A.B., Macedo, G.C., dos Santos, R.W., Lobosco, M.: Implementation of a computational model of the innate immune system. In: ICARIS. (2011) 95–107
5. Goutelle, S., Maurin, M., Rougier, F., Barbaut, X., Bourguignon, L., Ducher, M., Maire, P.: The hill equation: a review of its capabilities in pharmacological modelling. Fundamental & clinical pharmacology **22**(6) (2008) 633–648
6. Wagner, J.G.: Kinetics of pharmacologic response i. proposed relationships between response and drug concentration in the intact animal and man. Journal of Theoretical Biology **20**(2) (1968) 173 – 201
7. Price, T., Ochs, H., Gershoni-Baruch, R., Harlan, J., Etzioni, A.: In vivo neutrophil and lymphocyte function studies in a patient with leukocyte adhesion deficiency type ii. Blood **84**(5) (1994) 1635–1639
8. Su, B., Zhou, W., Dorman, K.S., Jones, D.E.: Mathematical modelling of immune response in tissues. Computational and Mathematical Methods in Medicine: An Interdisciplinary Journal of Mathematical, Theoretical and Clinical Aspects of Medicine **10** (2009) 1748–6718
9. Felder, S., Kam, Z.: Human neutrophil motility: Time-dependent three-dimensional shape and granule diffusion. Cell Motility and the Cytoskeleton **28**(4) (1994) 285–302
10. Chettibi, S., Lawrence, A., Young, J., Lawrence, P., Stevenson, R.: Dispersive locomotion of human neutrophils in response to a steroid-induced factor from monocytes. J Cell Sci **107**(11) (1994) 3173–3181