ISSN 0103-9741

Monografias em Ciência da Computação
n° 18/08

Cell Simulation: an Agent-based Software Engineering Approach

Maíra Athanázio de Cerqueira Gatti
Carlos José Pereira de Lucena

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 18/08 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena April, 2008

Cell Simulation: An Agent-based Software Engineering Approach

Maíra Athanázio de Cerqueira Gatti and Carlos José Pereira de Lucena

Laboratório de Engenharia de Software – LES
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil

{mgatti, lucena}@inf.puc-rio.br

Abstract. The simulation of cellular processes involves diverse components and complex interactions. All cellular subsystems are highly nonlinear, and subsystem couplings are often nonlinear as well. This nonlinearity indicates that the whole system is not equivalent to the sum of its subsystems and their interactions will produce emergent phenomena. We also know that multi-agent systems are heterogeneous interactive systems composed of subsystems called agents which produce an emergent behavior. It is our belief that to successfully model such cell processes, simulation systems must meet a number of computational requirements, from the area of agent-based software engineering in its design and implementation. That said, this paper argues why we need an agent-based software engineering approach for modeling and simulating cell behavior and processes. To support our hypothesis we have developed a framework that can be reused for simulating different kinds of cells and different cellular processes rather than only stem cell behavior. There is a 3D visualization tool and the framework can be instantiated to different differentiation processes rather than only to neuron generation.

Keywords: Multi-agent Systems, Computational Modeling, Stem cells.

Resumo. A simulação de processos celulares envolve diversos componentes e interações complexas. Todos subsistemas celular são altamente não lineares, e acoplamentos de subsistemas são freqüentemente não lineares também. Esta não linearidade indica que o sistema como um todo não equivale a soma de duas partes, ou subsistemas, e suas interações produzem fenômeno emergente. Também sabemos que sistemas multiagentes são sistemas interativos heterogêneos compostos de subsistemas chamados agentes que produzem comportamento emergente. Acreditamos que para modelar de forma bem sucedida modelos tais como processos de células, sua simulação deve atender a um numero de requisitos computacionais da área de engenharia de software baseada em agentes no seu projeto e desenvolvimento. Desta forma, este artigo discute porque é necessário uma abordagem de engenharia de software baseada em agentes para a modelagem e simulação do comportamento celular e seus processos. Para apoiar nossa hipótese, desenvolvemos um framework que pode ser reutilizado para a simulação de diferentes tipos de células e diferentes processos celulares ao invés de somente comportamento celular de células-tronco, o que foi o propósito inicial do projeto. Desenvolvemos uma ferramenta de visualização 3D e o framework pode ser instanciado para diferentes processos de diferenciação celular sendo que instanciamos para a geração de neurônios.

Palavras-chave: Sistemas Multiagentes, Modelagem Computacional, Células-tronco.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br

Table of Contents

I. Introduction

The systems biology community is building increasingly complex models and simulations of cells and other bio-logical entities. Their main goal is to create a predictive model for a programmable cell that can be optimized for personalized therapy. They do analysis of signal transduction pathways, reverse engineering of bio-molecular regulatory networks and try to unravel hidden cellular dynamics.

This community is beginning to look at alternatives to traditional representations, such as those provided by ordinary differential equations (ODE) (e.g., [15], [16]) or cellular automata (e.g., [17]-[19]). Differential equations are quite distant from the language of physicians. To consider computational models based on cellular automata, it would be necessary to know how to handle all possible interactions, which in most situations it is not possible because emergent behavior appears over time from the interactions. Moreover, the model has to consistently ex-plain the broad variety of experimental observations.

It is our intention to link these macroscopic phenomena to the underlying (latent) microscopic mechanisms. Also to include into the analysis experimental observations, which describe individual cell behavior, the model must be able to describe single cells as well as cell population behavior. Moreover, the ability to predict system behavior with a model helps evaluate model completeness as well as to improve our understanding of the mechanisms of biological processes.

Systems, which exhibit characteristics such as autonomy, pro-activity, interactivity and adaptation, might be modeled and engineered as multi-agent systems. For instance, not only are biological systems an excellent application area for multi-agent systems concepts and development technologies; as they reciprocally inspire new models for new software phenomena as self-adaptation, self-protection, self-healing, heterogeneity, self-organization, cooperation and coordination mechanisms (for instance, see [72]-[74]). By inspiring we mean that it is possible to apply the knowledge obtained from the study of biological systems to contribute to innovation in the engineering of multi-agent systems.

Much effort has been invested on the development of appropriate software engineering methods and technologies for multi-agent systems in the last few years [20], [21], [65]-[69]. Several methodologies, frameworks and platforms have been developed and proposed seeking to support software engineers in multi-agent system development: from coordination of multi-agent systems [22] to several strategies of negotiation between agents [23], [24], for instance.

With respect to the medicine point of view, if the model is sufficiently detailed and accurate, it serves as a reference, a guide for interpreting experimental results and a powerful means of suggesting new hypotheses. Moreover, the simulation lets physicians test experimentally unfeasible scenarios and can potentially reduce experimental costs and time (experiments in vitro can last weeks; the same experiments can take just a few minutes if they are done in silico). Besides, for instance, such a technique avoids ethical problems in the stem cell re-search regarding the use of embryonic stem cells.

Models are simplifications, abstractions. They are a crucial means of relating theory to reality and there is wide agreement that we must at least dimension them to relate to that reality and the theory on which they are based. On the other hand, agent-based models (ABM) or multi-agent systems (MASs) usually deal with objects that demonstrate some sort of dynamic and autonomous behavior.

Usually agents or objects react either passively to their environment or to other agents, or actively to their environment or other agents. In this sense, agents usually move in spatial systems like cities, which means motion or movement. Usually, there are many agents — not one or two but n. When there are many agents, they react to each other through time and their collective behavior can be unpredictable, surprising, hence novel and emergent. In this way, this style of modeling is quite consistent with the sciences of complexity.

An agent-based simulation (ABS) is a simulation with many intelligent agents interacting among themselves and with the environment. In a typical ABS of social behavior, the agents are the individuals that take rational decisions based on their neighbors' decisions. Very interesting social phenomena have been recently investigated, such as, for example, cooperation, social instability, and crowd modeling.

The great advantage of this modeling technique is that the emergent phenomena can be modeled through very simple rules governing the behavior of each agent. The global effect resulting from the interaction of the individuals is often unpredictable. To summarize, the point we want to make in this paper is that, an agent is a high-level software abstraction that provides a convenient and powerful way to describe a complex software entity in terms of its behavior within a contextual computational environment. Furthermore, the dynamic structures pre-sent in biological systems can be intuitively represented and efficiently implemented in agent-oriented simulators.

To support our hypothesis we developed a multi-agent framework [75], [76] for the cell simulation. On top of this framework we developed the stem cell behavior framework [78]. Those frameworks allow the reuse of modeling and design of different cells types within those entities and different cellular process and also the reuse of the stem cell processes modeling, which can be instantiated to different differentiation processes rather than only to neuron generation that was our intended instance.

That said, we present in section 2 the essence of cellular processes modeling and simulation with non software engineering computational perspectives so the reader can be contextualized on the kind of work to be developed. In section 3 we present de motivation, adequacy and advantages of an agent-oriented software engineering approach compared to non-agent-based related work. Afterwards we detail our agent-based software engineering approach to tackle this field through an exemplar application we have been working during the last year: the agent-based stem cell behavior simulation. Finally we end the paper with some discussions, final remarks and some future research work.

II. The Essence of Cellular Processes Modeling and Simulation

The theoretical and practical bases of simulating metabolic pathways are well grounded [63]. However, the design and implementation of simulation software and model-construction methods, covered by this paper, are still under active discussion.

Many attempts have been made to simulate molecular processes in both cellular and viral systems. Perhaps the most active area of cellular simulation is the kinetics of biochemical metabolic pathways. Several software pack-ages for quantitative simulation of biochemical metabolic pathways, based on numerical integration of rate equations, have been developed, including GEPASI [61], KIN-SIM [48], [51], MIST [52], METAMODEL [62], SCAMP [59] and E-CELL [64].

In predicting cell behavior, the simulation of a single or a few interconnected pathways can be useful when the pathway(s) being studied is relatively isolated from other biochemical processes. However, in reality, even the simplest and most well-studied pathways, such as glycolysis, can exhibit complex behavior due to connectivity. Moreover, simulations of metabolic pathways alone cannot account for the longer time-scale effects of processes such as gene regulation, cell division cycle and signal transduction.

Several groups have proposed and analyzed gene regulation and expression models by simulation [57], [54]. The cell division cycle [56], [60] and signal transduction mechanisms [58] have also been active areas of research for biological modeling and simulation. Most of them have utilized qualitative models to deal with the general lack of quantitative data in molecular biology. However, while qualitative models are generally useful when in-formation is incomplete [49], [50], they often generate ambiguous results [55], the behaviors of which are difficult to predict due to combinatorial explosion (for a re-view on computer simulations in biology, see [53]).

Previous studies in biochemical and genetic simulations have usually limited their models to focus on only one of the several levels of the time-scale hierarchy in cellular processes. Linking the gaps between the various levels of this hierarchy is an extremely challenging problem that has yet to be adequately addressed. This paper presents a step towards integrative simulation of multi scale cellular processes.

III. A Case for an Agent-Based Software Engineering Approach

To understand cellular systems, it is necessary to step behind and understand biology systems. Biology is the study of complex adaptive reproducing systems. Systems biology is the quantitative study of biological systems, aided (or hindered) by technological advances that permit computational analysis of observations [25].

A biological system, understood as a computational sys-tem, represents computational units that might be interpreted, on different levels of abstraction, as proteins, cells, tissues, organs, etc.) running in parallel (following well-defined patterns of behavior determined by the potential bio-chemical reactions in which they might be involved) and organized in hierarchies of subsystems an organism can be described as a system of organs, then each organ as a system of tissues and further the tissues as systems of cells, etc.). They interact, collaborate, communicate and interrupt each other. Underlying this paradigm is the assumption that each part of such a system (each subsystem) has its own identity, which persists through time [36].

By abstracting biological systems on the level of their behavior, we obtain behavioral models that share many characteristics with computational systems. Thus we have concurrency, event-driven and cause-effect behaviors and branching-time dependence, all in the context of distributed control [36]. Biological systems are complex [37], consisting of a set of components interacting with each other and with an external (dynamic) environment. Here we summarize the main multiagents' features [6] related to complex systems – hence biological systems:

1. Multiagents systems interactions are non-linear: a small perturbation may cause a large effect, a proportional effect, or even no effect at all; in biological systems, for instance, we have the biochemical and cellular rhythms or oscillations.

2. Multiagents systems contain feedback loops: the effects of an element's behavior are fed back in such a way that the element itself is altered;

3. Multiagents systems can be designed as open systems, in biological systems matter or energy can flow into and/or out of the system;

4. Agents and multiagents systems can learn, so they have a memory; and biological systems are dynamical systems that change over time, and prior states may have an influence on present states; they must be able to newly activate a previously performed reaction.

5. Agents may be nested: the components of a multi-agents system may themselves be multiagents systems, and for instance, many biological systems are described hierarchically as components of sub-systems;

6. Multi-agent systems provide flexibility for modeling more sophisticated, globally emergent behavior: the global effect resulting from the interaction of the agents is often unpredictable and non-deterministic; In the same way biological systems may produce emergent phenomena [38]-[46], which can be seen as an evolving process that leads to the creation of novel coherent structures, patterns of behavior, and properties at the macro level that dynamically arise from the interactions between the parts at the micro level. There are simply no levers that can be pulled in order to produce a particular kind of emergent result.

7. Agents are autonomous and interactive entities: an agent is capable of acting without direct external intervention and communicates with the environment and other agents, hence multi-agent systems are capable of being self-organized: agents could be organized in a structure that might evolve to a different structure according to the agent's behavior, performance, and others. In the same way bio-logical systems are self-organizing systems: dynamical and adaptive systems functioning without external direction, control, manipulation, interference, pressures or involvement [39], [47].

8. Multi-agent systems can be orchestrated in order to demonstrate a coordination mechanism and biological systems have coordination mechanisms, for instance, based on the specialization of certain cells, which will become able to interact and activate their specific working when activated by the direct interaction with other entities with compatible membrane;

9. Agents and multi-agents systems have the capacity for adaptation: an agent is capable of responding to other agents and/or its environment to some degree, and a multi-agent system might adapt itself to a specific state through the learning processes; in biological systems we have a fascinating adaptation phenomena represented by the human body immune system: there is no prior knowledge about possible threats.

Just to clarify, self-organization and emergence have some similarities and some differences. They are both self-sustained systems that are not directly controlled or manipulated in any way from the outside. They both evolve over time; however, only self-organizing systems need to exhibit a goal-directed development. Emergent systems consist of a larger number of low-level (micro-) entities, which collaborate in order to exhibit a higher level (macro-) behavior. The unavailability of one or more of those lower level entities does not abrogate the functioning of the system (graceful degradation) while this may be the case in self-organizing systems.

To sum up, multi-agent systems provide abstractions that allow decomposing a biological system to a set of agents; provide flexibility for modeling more sophisticated, glob-ally emergent behavior: the global effect resulting from the interaction of the individuals is often unpredictable and non-deterministic; Multi-agent systems by their nature are powerful tool for modeling biological systems [5]. Cellular (biological) systems are complex systems and their modeling implies a deep understanding of the system both in terms of its structure and its behavior and multi-agent systems allows this specification. Software agents embody distribution and heterogeneity and, thus, they are indicated as the new abstraction for the engineering of complex distributed systems.

That said, considering that the underlying mechanisms and the regulatory principles of cellular organization are still widely unknown and that they are self-organizing system that shows emergent behavior, MAS is an effective way to understand cellular organizations, and to deal with it emergent global behavior. Moreover, the agent-based simulation suggests how tiny changes in individual stem cell behavior might lead to disease at the global through the emergent behavior, allows temporal analysis, reduce costs and risks, and could avoid some ethical is-sues.

III.i. Advantages of Multi-Agent Systems compared to Non-Agent-Based Related Work

None of the mathematical models (for instance, see [16] for differential equations) used for describing biological systems allow expression of partial information about a system, i.e. to formally describe open systems. Moreover, depending on the system's complexity, there would be an explosion of differential equations; for example, to model it with more than 50 equations to model a subsystem. Another drawback is the absence of an abstraction for the models. Physicians must deeply understand mathematic-cal methods in order to model the system, while multi-agent systems can provide the right level of abstraction for that.

Compared to the biological systems modeling and simulation using Monte Carlo [26] methods, such as [27]-[29], multi-agent systems are not just probabilistic dependent. More than reproducing the emergent behavior, they can provide advanced mechanisms existent in biological systems, such as learning and adaptation that, as far as we know, are not possible to implement through Monte Carlo simulation. Those mechanisms not only make the model more complete but also allow the optimization of self-organization, for instance.

Considering the cellular automata approach [30] on which several biological systems have been modeled and simulated (e.g., [31][32]), the multi-agent system approach for modeling and simulating biological systems might be more suitable since it provides an easier way of represent the interactions between entities through the agents' interactions. Moreover, the software engineering for multi-agent systems can provide powerful techniques, methods and tools for the engineering of modeling and simulation of biological systems. For instance, self-organization of biological systems could be modeled through the self-organization modeling techniques existing in agent-oriented methodologies that accomplish this purpose.

Addressing the Petri Nets [33], [34] approach for modeling biological systems [35], they are not suitable for studying systems exhibiting continuous dynamic behavior that: (1) cannot be described by a set of discrete states, (2) cannot be broken down to atomic processes, or (3) are dependent on spatial properties. Examples include fluid dynamics and protein folding. And multi-agent systems could address all of these different kinds of behaviors.

The MAS model is a powerful tool used to describe lo-cal behavior and leaves the system free to simulate all events just by interactions between agents. On the other hand, it important to highlight that the not-agent-based related work cited are powerful ways of modeling and simulating biological systems and have been proven to work. Instead, we need understand how multi-agent systems complement these approaches in nature and behavior.

IV. An Exemplar Application: The Agent-Based Stem Cell Behavior Simulation

We have been doing research on an exemplar application for demonstrating the multi-agent system software engineering applicability for cell simulation. In this section we present the exemplar application chosen: the stem cell behavior.

Stem cells play a prominent role in biology and life sciences. Their importance is growing more and more, not only in basic research fields such as cell or developmental biology, but also in medicine and clinical research. The main reason underlying this broad interest in stem cells is their capacity to reconstitute functional tissues after disturbance or injury. They are able to produce a huge number of differentiated, functional cells and, at the same time, they maintain or even re-establish their own population.

Basically, stem cells are a potentially heterogeneous population of functionally undifferentiated cells; com-posed of multi-cellular organisms; capable of homing to an appropriate growth heterogeneous environment, proliferation, production of a large number of differentiated progeny, self-renewing or self-maintaining their population, regenerating the functional tissue after injury with flexibility and reversibility in the use of these options [8].

A stem cell is a primitive cell that can either self-renew (reproduce itself) or give rise to more specialized cell types. The new perspective on stem cell systems as net-works of different cell types and their interactions implies that stem ness should not be treated as an explicit cellular property, but as the result of a dynamic self-organization process. The microenvironment interactions and their specific effects on proliferation and differentiation have to be embedded in the concept.

Nowadays, stem cells are cultivated in the lab in order to differentiate into a specific mature cell. Today it is hard to predict the stem cell behavior under some substances. Moreover, the entire infrastructure necessary to maintain a stem cell culture is very expensive and many stem cells are wasted if the injected substance does not lead the culture to the desired mature cell. Thus, stem cell simulation is a powerful tool for reducing such costs and accelerating the stem cell therapy process.

To date d'Inverno et al. [13], [1]-[3] have taken the first steps towards an agent-based software engineering for the stem cell modeling and simulation. They produced formal and mutually consistent specifications of the leading of some predictive models of stem cell behavior within their agent framework. They have also produced simulations of these models. In their approach, each stem cell is implemented as an agent. They modeled and simulate the stem cells in a dynamic environment with the capabilities of division and differentiation (stem cells which have reached their cycle phase and which are surrounded by stem cells become differentiated). However, the stem cell behavior modeled was too simple considering that they use probabilistic determinism instead of the emergent behavior raised from the signaling transduction pathways.

Therefore they can only achieve partial self-organization since they do not take in account that full self-organization emerges from intracellular interactions. Self-organization plays here a fundamental key regarding the validation process since stem cells are self-organized systems. They also argue that by building a formal model using a specification language from software engineering (they used the language Z [70]), there are techniques to ensure that the simulation correctly implements the model. However, constructing a formal model and correctness proof of a complex interacting computing system is infeasible [71] since one cannot model all possible behavior of an interaction model and thus formally proving correctness is not merely difficult but impossible.

IV.i. Our Agent-based Software Engineering Approach

Our agent-based software engineering approach for building cell simulations is briefly depicted in Fig. 1. First, we build qualitative models (such as pathway maps) from in vivo and in vitro data and hypotheses, or a reference model (qualitative modeling). This first phase occurs during the requirement analysis (considering the main software engineering phases) and must be intensively done by the molecular biologists, or for instance, the stem cell researchers. Then, quantitative characterization of cellular properties facilitates the transition to an agent-based system model (quantitative modeling). We then translate the numerical and dynamic properties of the quantitative model into an agent-based modeling language, we implement it (cell programming), and predict the systemic behavior (run). Analysis of the results suggests new hypotheses (analysis and interpretation), so it is possible to simulate the efficacy and safety demonstration and also methods to prevent rejection in the human body. Having the analysis we can be subsequently test by wet experiments, and the cycle begins anew until the specialists demonstrate the efficacy and safety in vitro so they can do human trials.

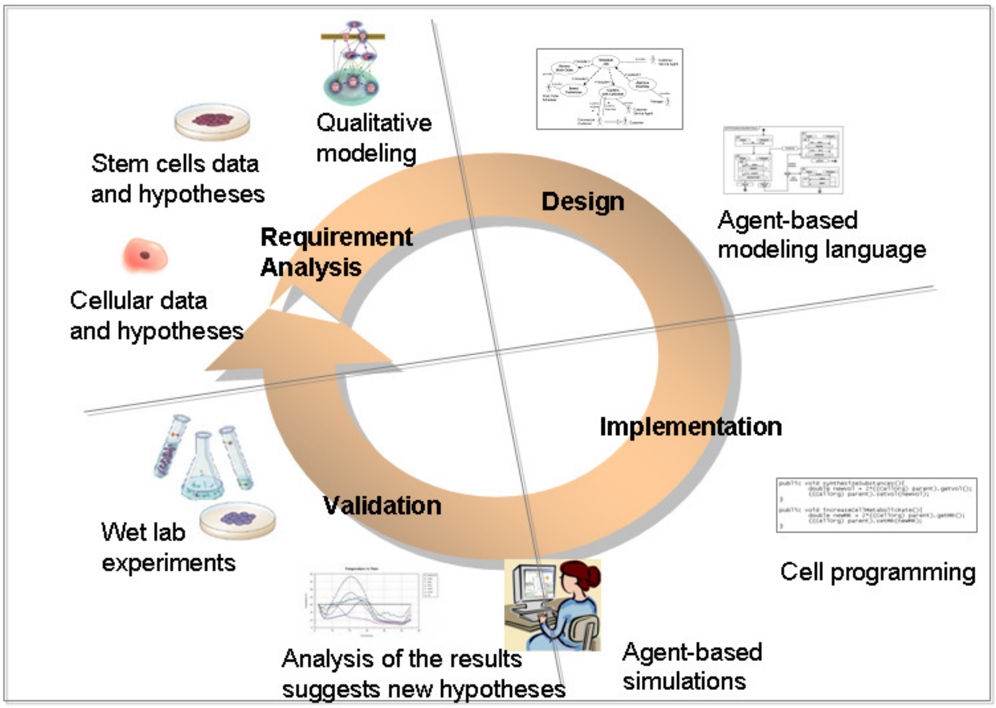

Qualitative modeling
Stem cells data and hypotheses
Design
Agent-based modeling language
Requirement Analysis
Cellular data and hypotheses
Implementation
Validation
Wet lab experiments
Cell programming
Analysis of the results suggests new hypotheses
Agent-based simulations

Fig. 1. The wet lab process enhanced with the agent-based approach.

We developed an agent-based solution to model and simulate the stem cell processes and the internal cell life-cycle . Our agent-based cell computational model is a distributed autonomous entity composed of reactive and pro-active agents. It can perceive and signal the environment. To determine how cells behave and interact, we need to understand how information is transferred among and within cells and how it changes the environment and other cells states (cellular data and hypotheses, and qualitative modeling). In the stem cell systems, an important entity with an active influence in the process is the Niche. The Niche is a specialized cellular environment, which provides stem cells with the support needed for self-renewal, and contains the cells and proteins that constitute the extra cellular environment. The Niche has regulatory mechanisms in order to save stem cells from depletion and to protect the host from over-exuberant stem-cell proliferation [12]. The niche is the stem cells' habitat, as the environment is the agents' habitat.

At the agent-based stem cell computational model, there are four kinds of cells: multi-potent cells are cells with a full power of differentiation, that can give rise to several other cell types; precursor cells are cells there are able to self-differentiate into a specific kind of cell, for instance, a blood cell; progenitor cells are stem cells that have developed to the stage where they are committed to forming a particular kind of new specific cell; differentiated cells are specialized cells with no power of differentiation. In the case that the differentiation process generates a neuron we have: a stem cell like a multi-potent cell; a precursor neuron like a pre-cursor cell; a progenitor neuron like a progenitor cell and a neuron like a differentiated cell - for instance, a neuroblast and the neuron itself, respectively. In order to ensure self-renewal and differentiation, the stem cells undergo two types of cell division: symmetric division: giving rise to two identical daughter cells, both endowed with stem cell properties; and asymmetric division: produces only one stem cell and a progenitor or precursor cell with limited self-renewal potential until the mature cell generation with no differentiation potentiality.

In order to simplify the model we considered only the more actives components during the cell life-cycle and the mitosis division (which is the stem cell division during the self renew process). By actives components we mean components which contribute and influence more directly the cellular differentiation during the cycle. Cells, proteins, DNA and complexes such CDKs-Cyclins are agents. The cells can be a proliferative cell or non-proliferative cell. The stem, progenitor and precursor cells are proliferative, while the neuron cells are non-proliferative. CDK, Retinoic Acid, Cyclin and LIF are proteins.

The Cdk-Cyc Complex is the join between a CDK and a Cyclin protein. The cells can die or start a cell cycle phase. Each cell cycle phase may only be started if some events occur, this is the cell checkpoint mechanism For instance, during the prometaphase phase the cell checkpoint checks if all the kinetochore micro-tubules were attached to each chromosome, and if the nuclear membrane was dissolved in order to go to the metaphase phase. The DNA which is an adaptive agent interacts with the proteins during the molecular pathways regulation.

All the cell life-cycle concepts were implemented in a multi-agent-based framework for the cell simulation. On top of this framework we developed the stem cell behavior framework [78]. We needed a framework because there are different kinds of stem cells providing for a broad spectrum of proliferated progenitor cells that ultimately lead to the creation of every adult cell type necessary for sustaining life. Whether it be a hematopoietic stem cell giving rise to all of the cell types typically found in whole blood (Red blood cells, platelets, leukocytes such as neutrophils, eosinophils and basophils as well as monocytes (macrophages), and T and B cell lymphocytes), or a neural stem cell which can give rise to neurons, astrocytes, and oli-godendrocytes (Fig. 2), all stem cells are common with respect to their primitive nature and they ideally exist without any signs or markers of differentiation and lineage commitment.

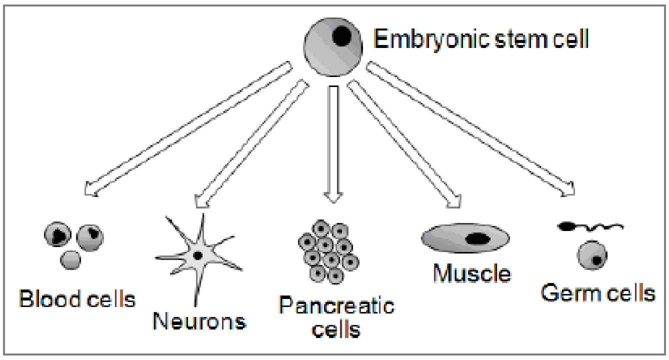


Fig. 2. Several kinds of cells that the stem cell can be differentiated into.

Those frameworks allow the reuse of structural relationships and dynamic interactions modeling and de-sign of different cells types within those entities and different cellular process and also the reuse of the stem cell processes modeling, which can be instantiated to different differentiation processes (Fig. 3) rather than only to neuron generation which is our instance. It also allows us to integrate different signal transduction pathways, and speed the reverse engineering of bio-molecular regulatory networks through those integrations.

Basically, the framework uses four design patterns [4]. The pattern Prototype is used to implement the stem cell's substances because this facilitates the objects' clone process. The pattern State is used to implement the states the cell can be in. The pattern Strategy is used to implement the cell's asymmetric division strategy. Each type of cell has one distinct asymmetric division strategy. The singleton is used to implement the niche, because the niche is unique.

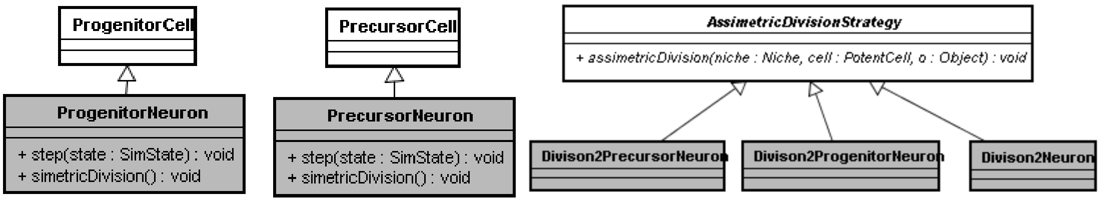

ProgenitorCell
ProgenitorNeuron
+ step(state : SimState) : void
+ simetricDivision() : void
PrecursorCell
PrecursorNeuron
+ step(state : SimState) : void
+ simetricDivision() : void
AssimetricDivisionStrategy
+ assimetricDivision(niche : Niche, cell : PotentCell, o : Object) : void
Divison2PrecursorNeuron
Divison2ProgenitorNeuron
Divison2Neuron

Fig. 3. The cells and asymmetric division hot spots (partial view).

Briefly, the first hot spot that must be implemented is the environment where the cells represented by the agents live: the niche. Other important hot spots are the progenitor and precursor cells, which must be instantiated depending on the process for a specific type of cell to be simulated. It is necessary to add features and behaviors depending on the type of cells to be instantiated. The strategy for asymmetric divisions from stem cells, progenitor (neurons) and precursor (neurons) cells also must be instantiated.

Moreover, if we will simulate we have to spend some effort on appropriate ways of visualizing the simulation in multi scale. The real life, 3d spatial self-organization of stem cells developed [79] has the goal to show the embryonic body formation from an initial number of stem cells. It is roughly explained as follows. The stem cells divide itself into two news cells, due to the mitoses division process (Fig. 4). Both cells assume new positions: one assumes the parent position and the other an adjacency position. Before the division process, the cell grows up, pushing its neighbors away and occupying the required space for the new cell.

The development of a stem cell behavior simulator has also as objectives: offer to the users a visual environment by means of which is possible to follow the Macro and Micro levels of the simulation of stem cell's cellular life cycle in the niche; perceive the difficulties of implementation of the proposed model; validate the model, observing if the simulated behavior has similarities with the behavior of the real entities (stem cell's); and test several scenarios.

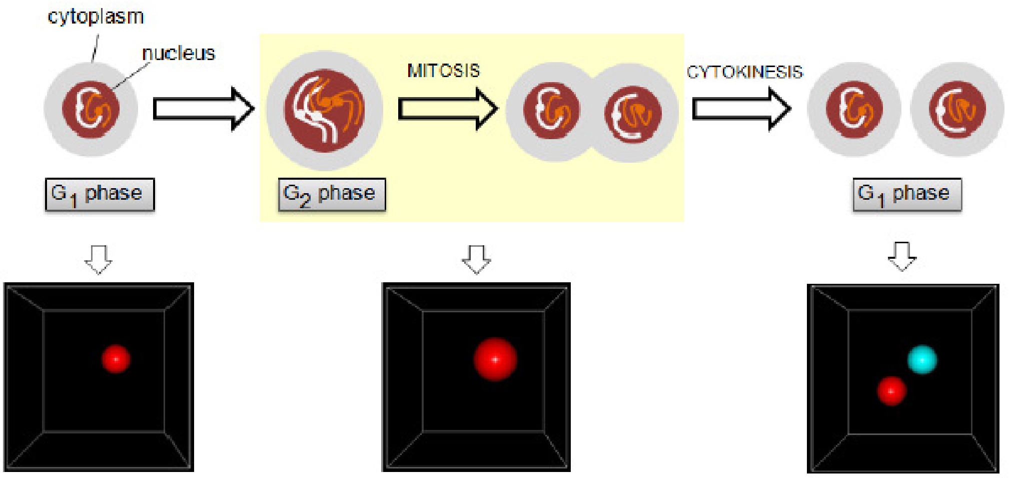
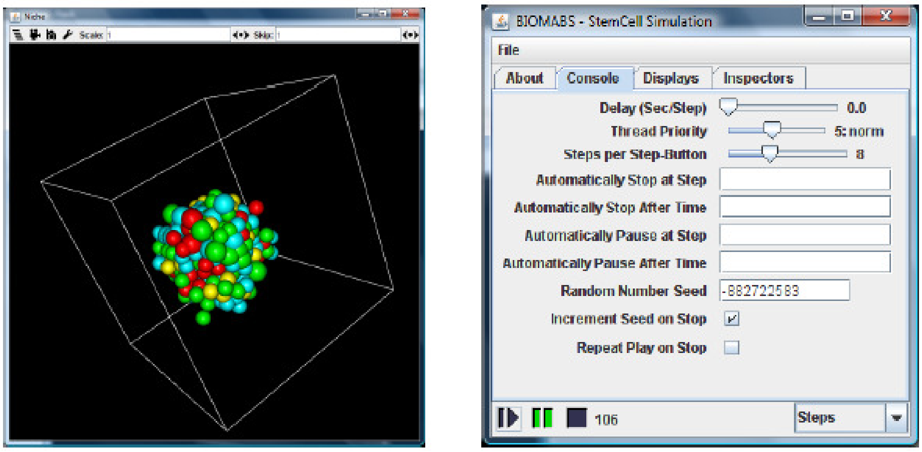


Fig. 4. Division process: on the left, the figures represent the division steps in real life; on the right the figures are their representation in the system.

We understand as Macro scale the emergent behavior proceeding of the interactions between the simulated entities. This scale is presented by the simulator to the users by means of a visualization area (3D) that represents the niche where the cells evolve in their life-cycles. In another scale, each phase of cell life-cycle has a 2D graphical representation, presenting the state of the main entities involved in process. These graphical representations, besides presenting a phase of the life-cycle differentiate by means of colors the capacity of differentiation of the cell. As Micro scale we under-stand the state and behavior of each entity simulated individually. In the simulator tool it is possible to obtain the micro scale through the internal state of the cell selected and chronologically each internal interactions occurred by the selected cell.



Delay (Sec/Step) 0.0
Thread Priority 5: norm
Steps per Step-Button 8
Automatically Stop at Step
Automatically Stop After Time
Automatically Pause at Step
Automatically Pause After Time
Random Number Seed -882722583
Increment Seed on Stop ☑
Repeat Play on Stop □
106

Fig. 5. Screenshot of the user interface of the system: it shows the display window and, it shows the console window.

Fig. 5 shows a snapshot of the interface of the frame-work instance during a simulation. The Macro scale, emergent behavior can be seen in the visualization area. And in the console window, to date, the specialists might speed, pause, play, stop or update some simulation settings.

V. Discussion and Final Remarks

We have argued why we need an agent-based soft-ware engineering approach for modeling and simulating cell behavior and processes since an agent is a high-level software abstraction that provides a convenient and powerful way to describe a complex software entity in terms of its behavior within a contextual computational environment. The great advantage of this modeling technique is that the emergent phenomena can be modeled through very simple rules governing the behavior of each agent. The global effect resulting from the interaction of the individuals is often unpredictable. Furthermore, the dynamic structures pre-sent in biological systems can be intuitively represented and efficiently implemented in agent-oriented simulators.

To support our hypothesis we developed a framework that can be reused for simulating different kinds of cells and different cellular process rather than only stem cell behavior. Not only we can reuse all the modeled already developed but we can model learning, adaptive behavior and open systems. There is a 3D visualization tool and the framework can be instantiated to different differentiation processes rather than only to neuron generation.

The stem cell researchers' collaborators were very excited with the first results. Basically, they observed in the visualization of the neuron instance the first emergent phenomenon that is similar to the emergent phenomenon in vitro: the differentiated cells are located at the embryonic body's extremity while the specialized and stem cells are located at the embryonic body's center.

Although the number of cells running together and the time of execution in the simulation were satisfactory, we need to increase this number and to achieve this goal we are distributing the framework and application in a grid architecture with ten processors. If we have around 20,000 cells we can reach more refined self-organizing mechanisms that might occurs in those kinds of systems.

Moreover, the next steps are a review of a set of tests on the proposed model, intending to adjust it for the demand of the biomedical domain. Other activities are to produce reports about Macro and Micro scale to support the comprehension of the information visualized during the simulation process, and identify with the specialists of the domain the trustworthiness of the results achieved through this first instance, in order to allow an adaptation and bias adequate to the reality of the research of the domain. Study of each protein and substances and how these relate must be carried out and reproduced in this application.

Besides the indicated future work, an important fundamental engineering issue is to achieve a macroscopic behavior that meets the requirements and emerges only from the behavior of locally interacting agents when designing self-organizing emergent multi-agent systems. To date, agent-oriented methodologies are mainly focused on engineering the microscopic issues, i.e. the agents, their rules, the protocols, how they interact, etc, without explicit support for engineering the required macroscopic behavior. As a consequence, the macroscopic behavior is achieved in an ad-hoc manner. A fundamental problem is the lack of a method that allows us to systematically specify desirable macroscopic properties, map them to the behavior of individual agents, build the system, and validate the required macroscopic properties.

So we are working on a method and a representation model to handle this issue. However, not only are bio-logical systems an excellent application area for multi-agent systems concepts and development technologies, as they reciprocally inspire the representation model that accomplishes the design requirements for modeling cellular processes. In general complex systems and new models for software phenomena as the case of Autonomic Computing: self-adaptation, self-protection, self-healing, heterogeneity, self-organization, and also cooperation and coordination mechanisms. For short, it is possible to apply the knowledge obtained from the study of biological systems with new concepts for the design of robust self-organized multi-agent systems using ideas inspired by molecular and cellular systems biology.

Also, as a future work, we plan to optimize the emerging behavior generated by the self-organizing stem-cell represented by agents. By optimization we mean the establishment of optimum differentiation or proliferation rates, for instance, through the addition and removal of some specific factors in the niche. Hence, the challenge would be to define macro proper-ties and, starting from local interactions, to integrate a specialized online search planner to optimize the behavior so that the macro properties can be satisfied. Therefore, the simulator might allow more interactions with the simulation environment increasing the tool usability and dependability, as well as helping the validation process vis-à-vis the in vitro process.

ACKNOWLEDGMENT. The authors wish to thank team of collaborators that participate in this project: Diego Bispo, Geisa Faustinho, Bruno Soares, and the stem cell researchers from LANDIC/UFRJ, which is the Neurogenesis and Cell Differentiation Lab from the Universidade Federal do Rio de Janeiro: professor Stevens Rehen and Priscila Britto.

References

[1] d'Inverno, M. and Prophet, J. (2004). Modelling, simulation and visualisation of adult stem cells. In P. Gonzalez, E. Merelli, and A. Omicini, editors, Fourth International Workshop on Network Tools and Applications, NETTAB, pages 105–116, 2004.

[2] d'Inverno, M. and Saunders, R. (2005). Agent-based modelling of stem cell organisation in a niche. In Engineering Self-Organising Systems, volume 3464 of LNAI. Springer, 2005.

[3] d'Inverno, M., Theise, N. D. and Prophet, J. (2005). Mathematical modelling of stem cells: a complexity primer for the stem cell biologist. In C. Potten, J. Watson, R. Clarke, and A. Renehan, ed., Tissue Stem Cells: Biology and Applications. Marcel Dekker.

[4] Emonet, T., C.M. Macal, M.J. North, C.E. (2005) Wickersham, and P. Cluzel, 'AgentCell: A Digital Single-Cell Assay for Bacterial Chemo-taxis,' Bioinformatics, Vol. 21, No. 11, pp. 2714-2721, Oxford University Press, Oxford, UK (March 17, 2005).

[5] Gatti, M.; Lucena, C. (2007). An Agent-Based Approach for Building Biological Systems: Improving The Software Engineering for Complex and Adaptative Multi-Agent Systems. Monografias de Ciências da Computação, PUC-Rio, Rio de Janeiro, v.14/07, pg. 1-22, 2007.

[6] Jennings, N., R. (2000). On Agent-Based Software Engineering. Artificial Intelligence Journal, 117 (2) 277-296, 2000.

[7] Lei, J., Mackey, M.C. (2007). "Stochastic differential delay equation, moment stability, and appplication to hematopoietic stem cell regulation system", SIAM J. Appl. Math. (2007), 67(2), 387–407.

[8] Loeffler, M. and Roeder, I.. Cells Tissues Organs, 171(1):8-26, 2002.

[9] Lord, B.I.; in Stem cells, Cambridge Academic Press, 1997, pp 401-422.

[10] Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. J. Amer. Stat. Assoc. 44, 335-341, 1949.

[11] Minar N., Burkhart R., Langton C. e Askenaa M. (1996) The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations, http://www.swarm.org.

[12] Scadden, D. T. (2006). The stem-cell niche as an entity of action. Nature 441, 1075-1079 (29 June 2006) | doi:10.1038/nature04957; Published online 28 June 2006.

[13] Theise, N. D. and d'Inverno, M. (2003). Understanding cell lineages as complex adaptive systems. Blood, Cells, Molecules and Diseases, 32:17–20, 2003.

[14] Wichmann, H. E. and Loeffler, M.; Mathematical modeling of cell proliferation: Stem cell regulation in hemopoiesis, CRC Press, Boca Raton, 1985.

[15] S. H. Kleinstein and P. E. Seiden. Simulating the Immune System. Com-puting in Science and Engineering, 69-77. July/August 2000.

[16] J. Gruber. Models of immune systems: The use of differential equations. http://www.lymenet.de/literatur/immundif.htm.

[17] S. Bandini. Hyper-cellular automata for the simulation of complex biological systems: a model for the immune system, in: D. Kirschner (Ed.), Special Issue on Advance in Mathematical Modeling of Biological Processes, Vol. 3, 1996.

[18] J. O. Kephart, G. B. Sorkin, M. Swimmer, S. R. White: Blueprint for a Computer Immune System, in: D. Dasgupta (Ed.), Artificial Immune Systems and Their Applications, Springer{Verlag, pp. 221,241}, 1999.

[19] Z. Agur, Y. Daniel and Y. Ginosar. The universal properties of stem cells as pinpointed by a simple discrete model. Jour. Math. Biol. 2002; 44(1):79-86.

[20] F. Zambonelli and A. Omicini. Challenges and research directions in agent-oriented software engineering. Autonomous Agents and Multi-Agent Systems, 9(3):253--283, Nov. 2004.

[21] N. R. Jennings. On Agent-Based Software Engineering. Artificial Intelligence Journal, 117 (2) 277-296, 2000.

[22] M. Cremonini, A. Omicini, F. Zambonelli. Coordination and Access Control in Open Distributed Agent Systems: The TuCSoN Approach. Proceedings of the 4th International Conference on Coordination Languages and Models, p.99-114, September 11-13, 2000.

[23] S. S. Fatima, M. Wooldridge and N. R. Jennings. Approximate and online multi-issue negotiation. Proc. 6th Int. J. Conference on Autonomous Agents and Multi-agent Systems, Hawaii, USA, 2007.

[24] I. Rahwan, L. Sonenberg, N. R. Jennings and P. McBurney. STRATUM: A methodology for designing heuristic agent negotiation strategies. Int. J. of Applied Artificial Intelligence 21, 2007.

[25] Z. Szallasi, V. Periwal, J. Stelling. (Eds.). System Modeling in Cellular Biology: From Concepts to Nuts and Bolts. Cambridge, MA: MIT Press 2006.

[26] Metropolis, N. and Ulam, S.. The Monte Carlo Method. J. Amer. Stat. Assoc. 44, 335-341, 1949.

[27] T. D. Utteridge, D. E. Charlton, and B. J. Allen. Monte Carlo Modeling of the Effects of Injected Short-, Medium- and Longer-Range Alpha-Particle Emitters on Human Marrow at Various Ages. Radiat. Res. 156, 413–418 (2001).

[28] C. H. Turner, Monte Carlo simulation of equilibrium reactions at vapor-liquid interfaces. J Phys Chem B, vol. 109, 2005.

[29] J. Peter; W. Semmler. Integrating kinetic models for Simulating tumor growth in Monte Carlo Simulation of ECT systems. Nuclear Science, IEEE Transactions on , vol.51, no.5pp. 2628- 2633, Oct. 2004.

[30] S. Wolfram: Theory and Applications of Cellular Automata, World Press, 1986.

[31] B. Chopard and M., Cellular Automata Modeling of Physical Systems, Collection Aléa Saclay, Cambridge University Press, 1998. (ISBN 0-521-46168-5)

[32] S. Dormann and A. Deutsch. Modeling of self-organized avascular tumor growth with a hybrid cellular automaton. In Silico Biology, vol. 2, pp. 68-76, 2002.

[33] P.H. Starke. Analyse von Petri-Netz-Modellen. B. G. Teubner, Stuttgart, 1990.

[34] W. Reisig. Petri Nets: An Introduction. Springer, Berlin, 1985.

[35] M. Peleg, D. Rubin, and R. B. Altman. Using Petri Net Tools to Study Properties and Dynamics of Biological Systems. Journal of the American Medical Informatics Association 12(2):181-199, 2005.

[36] R. Mardare, C. Priami. A logic of observers for specifying Biological Systems. http://dit.unitn.it/~mardare/public_html/MeCBIC.pdf (last accessed in Feb. 2006)

[37] H Kitano. Foundations of Systems Biology. MIT Press, 2002.

[38] T. De Wolf, and T. Holvoet: Emergence and Self-Organization: A Statement of Similarities and Differences; Workshop on Engineering Self-Organizing Applications (ESOA); as part of Third Int. Joint Conf. on Autonomous Agents & Multi-Agent Systems AAMAS 2004, July 21-24, New York, USA; 2004.

[39] G. Nitschke: Emergent and Cooperative Behavior in Swarm Systems: State of the Art; Workshop on Engineering Self-Organizing Applications (ESOA); as part of

Third International Joint Conference on Autonomous Agents & Multi-Agent Systems AAMAS 2004, July 21-24, New York, USA; 2004.

[40] J. P. Crutchfield: Is Anything Ever New? Considering Emergence, SFI Series in the Sciences of Complexity XIX, Addison-Wesley 1994.

[41] J. Goldstein: Emergence as a Construct: History and Issues, Emergence, Vol 1, Issue 1, 1999.

[42] F. Heylighen: Self-organization, Emergence and the Architecture of Complexity, Proceedings of the first European Conference on System Science, Paris, 1989.

[43] P. Valckenaers et al.: On the design of emergent systems: an investiga-tion of integration and interoperability issues; Engineering Applications of Artificial Intelligence 16 (2003), pp. 377–393;

[44] S. Kauffman: At Home in the Universe: The Search for Laws of Com-plexity. Harmondsworth: Penguin, 1996.

[45] Seel, Richard: Complexity and Culture: New Perspectives on Organisational Change, Organisations & People, 7 (2), pp. 2-9. http://www.new-paradigm.co.uk/culturecomplex.htm, 2000.

[46] M. Ulieru,;R. Unland: Enabling Technologies for the Creation and Restructuring Process of Emergent enterprise alliances; International Jour-nal of Information Technology and Decision Making, World Scientific Publishing Co., Inc.; Vol. 3, No. 1 (March 2004), pp. 33-60

[47] H. Haken: Information and Self-Organization: A Macroscopic Approach to Complex Systems, Springer- Verlag, Germany, 1988.

[48] Barshop ,B.A., Wrenn,R.F. and Frieden,C. (1983) Analysis of numerical methods for computer simulation of kinetic processes: development of KINSIM-a flexible, portable system. Anal. Biochem., 130, 134-145.

[49] Bray ,D. (1998) SIGNALING COMPLEXES: Biophysical constraints on intracellular communication. Annu. Rev. Biophys. Biomol. Struct., 27, 59-75.

[50] Bray ,D., Bourret,R.B. and Simon,M.I. (1993) Computer simulation of the phosphorylation cascade controlling bacterial chemotaxis. Mol. Biol. Cell, 4, 469-482.

[51] Dang ,Q. and Frieden,C. (1997) New PC versions of the kinetic-simulation and fitting programs, KINSIM and FITSIM. Trends Bio-chem. Sci., 22, 317 MEDLINE Abstract

[52] Ehlde ,M. and Zacchi,G. (1995) MIST: a user-friendly metabolic simula-tor. Comput. Applic. Biosci., 11, 201-207.

[53] Galper ,A.R., Brutlag,D.L. and Millis,D.H. (1993) Knowledge-based simulation of DNA metabolism: prediction of action and envisionment of pathways. In Hunter,L. (ed.), Artificial Intelligence and Molecular Biology. AAAI Press/The MIT Press, CA/MA, pp. 429-436.

[54] Karp ,P.D. (1993) A qualitative biochemistry and its application to the regulation of the tryptophan operon. In Hunter,L. (ed.), Artificial Intelligence and Molecular Biology. AAAI Press/The MIT Press, CA/MA, pp. 289-324.

[55] Kuipers ,B. (1986) Qualitative simulation. Artif. Intell., 29, 289-338.

[56] McAdams ,H.H. and Shapiro,L. (1995) Circuit simulation of genetic networks. Science, 269, 650-656. MEDLINE Abstract

[57] Meyers ,S. and Friedland,P. (1984) Knowledge-based simulation of genetic regulation in bacteriophage lambda. Nucleic Acids Res., 12, 1-9. MEDLINE Abstract

[58] Novak ,B. and Tyson,J.J. (1995) Quantitative analysis of a molecular model of mitotic control in fission yeast. J. Theor. Biol., 173, 283-305.

[59] Sauro ,H.M. (1993) SCAMP: a general-purpose simulator and metabolic control analysis program. Comput. Applic. Biosci., 9, 441-450.

[60] Tyson ,J.J. (1991) Modeling the cell division cycle: cdc2 and cyclin inter-actions. Proc. Natl Acad. Sci. USA, 88, 7328-7332.

[61] P. Mendes, "Gepasi: A Software Package for Modeling the Dynamics, Steady States, and Control of Biochemical and Other Systems," Com-puter Applications in Biosciences (CABIOS), vol. 9, no. 5, Oct. 1993, pp. 563–571.

[62] Cornish-Bowden, A. and Hofmeyr, J.H., MetaModel: a program for modeling and control analysis of metabolic pathways on the IBM PC and compatibles, Comput. Appl. Biosci., 7:89–93, 1991.

[63] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Upper Saddle River, N.J., 1971.

[64] Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J.C., and Hutchison, C., Software environment for whole cell simulation, Bioinformatics, 15:72–84, 1999.

[65] Ricardo Choren, Alessandro F. Garcia, Holger Giese, Ho-fung Leung, Carlos José Pereira de Lucena, Alexander B. Romanovsky: Software Engineering for Multi-Agent Systems V, Research Issues and Practical Applications [the book is a result of SELMAS 2006]. Springer 2007

[66] Alessandro F. Garcia, Ricardo Choren, Carlos José Pereira de Lucena, Paolo Giorgini, Tom Holvoet, Alexander B. Romanovsky: Software Engineering for Multi-Agent Systems IV, Research Issues and Practical Applications [the book is a result of SELMAS 2005]. Springer 2006

[67] Ricardo Choren, Alessandro F. Garcia, Carlos José Pereira de Lucena, Alexander B. Romanovsky: Software Engineering for Multi-Agent Systems III, Research Issues and Practical Applications [the book is a result of SELMAS 2004]. Springer 2005

[68] Carlos José Pereira de Lucena, Alessandro F. Garcia, Alexander B. Romanovsky, Jaelson Castro, Paulo S. C. Alencar: Software Engineering for Multi-Agent

Systems II, Research Issues and Practical Applications [the book is a result of SELMAS 2003] Springer 2004

[69] Alessandro F. Garcia, Carlos José Pereira de Lucena, Franco Zam-bonelli, Andrea Omicini, Jaelson Castro: Software Engineering for Large-Scale Multi-Agent Systems, Research Issues and Practical Appli-cations [the book is a result of SELMAS 2002] Springer 2003.

[70] M. Spivey. The Z Notation (second edition). Prentice Hall International: Hemel Hempstead, England, 1992.

[71] T. De Wolf, T. Holvoet; Towards a Methodology for Engineering Self-Organising Emergent Systems; Self-Organization and Autonomic In-formatics (I), Volume 135 of Frontiers in Artificial Intelligence and Ap-plications; H. Czap et al. (Eds.); IOS Press, 2005.

[72] Kephart, Jeffrey and Chess, David M.(2003). The Vision of Autonomic Computing. IEEE Computer Magazine 36(1): 41-50.

[73] P. Lin, A. MacArthur, J. Leaney. Defining Autonomic Computing: A Software Engineering Perspective. IEEE. Proc. of the 2005 Australian Soft. Eng. Conf. (ASWEC'05).

[74] S. Bandini, F. Celada, S. Manzoni, G. Vizzari: Modelling the immune system: the case of situated cellular agents. Natural Computing 6(1): 19-32 (2007)

[75] SILVA, Viviane Torres da ; Lucena C. J. P. . Modeling Multi-Agent System. Communications of the ACM, v. 50, n. 5, p. 103-108, 2007.

[76] GARCIA, Alessando Fabricio ; Lucena C. J. P. ; COWAN, Donald D . Agents in Object-Oriented Software Engineering. Software: Practice and Experience, v. 34, n. 5, p. 489-521, 2004.

[77] Gatti, M., de Vasconcellos, J.E., Lucena, C.J.P.; An Agent Oriented Software Engineering Approach for the Adult Stem-Cell Modeling, Simulation and Visualization. Workshop SEAS, João Pessoa, 2007.

[78] D. B. Conceição, M. A. de C. Gatti and C.J.P. de Lucena. An Agent-based Framework for Stem Cell Behavior Modeling and Simulation. Monografias do Departamento de Informática, 2008.

[79] G. Faustino, M. A. de C. Gatti, C.J.P. de Lucena and M. Gattass. A Multi-Agent-based 3D Visualization of Stem Cell Behavior. Monografias do Departamento de Informática, 2008.