Multi-Agent Systems for Biomedical Simulation: Modeling Vascularization of Porous Scaffolds

Hamidreza Mehdizadeh[1], Arsun Artel[1], Eric M. Brey[1,2], and Ali Cinar[1]

[1] Illinois Institute of Technology, Chicago IL 60616, USA
{mehdizadeh,brey,cinar}@iit.edu, arsun.artel@mam.gov.tr
[2] Research Service, Hines VA Hospital, Hines IL 60141, USA

Abstract. An interesting application of multi-agent systems (MAS) is in modeling systems that can be represented by independent entities interacting together, the so-called agent-based modeling (ABM). In this paper MAS paradigm is used as a promising technique for representing complex biomedical systems. A brief survey of some ABM of biomedical systems is presented, followed by the description of a multi-layered agent-based framework developed in our own labs to model the process of sprouting angiogenesis (blood vessel formation) within polymeric porous scaffolds used for regenerative medicine. The ABM structure developed and challenges in modeling systems with a large number of rapidly increasing interacting agents are discussed. 2D and 3D case studies are presented to investigate the impact of scaffold pore structure on vessel growth. MAS provides a valuable tool for studying highly complex biological and biomedical systems, and for investigating ways of intervening in such systems.

Keywords: agent-based simulation, agent-based modeling, emergent behavior, angiogenesis.

1 Introduction

The number and diversity of the fields that use multi-agent systems (MAS) is rapidly growing. Computational biology and biomedical engineering communities have started using multi-agent modeling techniques to study complex systems with many interacting elements. During its relatively short but fast developing lifetime, agent-based modeling (ABM) paradigm has shown to be a promising way of representing biological and biomedical systems and describing their dynamic behavior.

The central idea in ABMs is to define agents that represent the building blocks of a system and to develop rules that regulate their interactions. The rules originate from the vast knowledge gained through many years of studying the individual components of these biological systems. Agent-based systems (ABS) are naturally suitable for modeling biological systems as they are comprised of individual discrete micro-scale constituents (e.g. cells) that interact with each other to form non-homogeneous macro-scale bodies (e.g. tissues and organs). Table 1

D. Kinny et al. (Eds.): PRIMA 2011, LNAI 7047, pp. 113–128, 2011.

provides an illustrative list of applications of ABM in biomedical engineering. A brief review highlights how ABM is used in developing biomedical models and displays the diversity of the applications.

Table 1. An illustrative list of biological and biomedical applications of ABM

Field	Example	Reference
Cancer	Multi-scale, multi-resolution brain cancer modeling	[27]
Tissue engineering	ABM of neovascularization within porous scaffolds	[3]
Angiogenesis	Module-based multiscale simulation of angiogenesis in skeletal muscle	[16]
Lung disease	ABM of inflammation and fibrosis following particulate exposure in the lung	[8]
Clinical	In silico experiments of existing and hypothetical cytokine-directed clinical trials using ABM	[1]
Morphogenesis	Simulating properties of in vitro epithelial cell morphogenesis	[13]
Bone	ABM of real-time signaling induced in osteocytic networks by mechanical stimuli	[4]
Epidemiology	ABM of the spread of the 1918-1919 flu in three Canadian fur trading communities	[19]

Zhang et al. developed a multi-scale agent-based framework for modeling cancerous brain tumors [27]. They simulated the growth and expansion of brain tumors, utilizing a novel multi-resolution approach to substantially reduce the computation time of the individual-based model while maintaining a comparably high predictive power. Artel et al. developed an agent-based model to investigate the effects of porous scaffold structure on rate and characteristics of angiogenesis (blood vessel formation) [3]. Liu et al. used an ABM with complex logical rules and equation-based models integrated into a uniform framework for studying angiogenesis in skeletal muscle [16].

Brown et al. developed an ABM to capture the features of inflammation caused by particulate exposure in the lung [8]. Their model considered a limited number of relevant interactions among different cell types and their effect on tissue damage. An used a simple ABM that could evaluate the dynamics of the innate immune response and demonstrated counterintuitive outcomes of the system and the difficulty of effective manipulation of complex multi-component systems [1]. Grant et al. developed a model that could capture the interconversion of different cell types, with behavior of cell agents being governed by a set of axioms derived from observation of real cell behaviors and a decision process embedded within each agent [13]. Ausk et al. developed an ABM for studying networks of bone cells when affected by mechanical stimuli [4].

O'Neil and Sattenspiel modeled the epidemic spread of the flu between three real communities by means of stochastic agent-based computer simulation [19]. Their model was successful in describing the course of the flu spread in small communities. Thorne et al. have reviewed biological ABMs and compared them with continuum computational models [24]. In another work, they emphasized the importance of combining ABMs with experimental work to gain new understanding from the experiments and presented a number of such models [25].

Recent advances in genetics and in molecular and cellular biology have generated a vast amount of biological information. A systematic approach is needed to integrate this information in a unified framework, and to facilitate its use for multi-scale modeling. Traditionally, the evolution and patterning of the naturally discrete cells were modeled using a set of continuous differential equations. Even simple ABMs can develop complicated behavioral patterns and provide valuable insight about the dynamics of real-world systems. The MAS paradigm provides a versatile modeling environment where each cell can be represented by a software agent that has a set of states and behaviors and an internal rule-based logic to relate them. A hierarchical MAS can link systematically the effect of mechanisms in a lower level of biological scale (e.g. molecular or cellular levels) with higher level (e.g. tissue level) patterns and behaviors [25]. The flexibility and modularity of MAS enable the development of a model that can be modified, extended, or expanded with a minimum amount of effort.

2 Use of Agents in Biological Systems

Agents are autonomous software entities designed to perform proactively or reactively various tasks related to the specific system they represent. In medical system simulation and modeling, agents usually represent cells or cellular components (such as cell receptors or integrins) that interact with each other and their local environment. These interactions lead to higher-level emergent phenomena, such as blood vessels or tissue formation, or cancerous tumor invasion.

A list of the characteristics of ABS that are important in simulating biomedical systems include:

1. *Modularity:* The behavior of different agents are determined at execution time based on rules and parameter values computed dynamically during the simulation. It is possible to add new rules at later stages of model development. It is also possible to add different types of agents to the model. The effects of new rules and agents on current rule-bases should be carefully considered for conflict resolution.
2. *Abstraction:* ABMs can be constructed with less quantitative information about the behavior of a system compared to models based exclusively on fundamental equations.
3. *Multi-scale modeling:* ABMs provide a powerful structure to describe phenomena occurring in different scales of biological systems in a single model. One model can include details in the molecular level, and produce results in the tissue or organ level.

4. *Randomness:* There are many unknowns in biological and biomedical systems due to lack of measurement or precision. By adding randomness to models, it is possible to accommodate the lack of knowledge and to account for the stochastic nature of phenomena at various physical or biological scales. ABM rules can represent the stochastic and deterministic phenomena.
5. *Decision-making at runtime:* An agent is capable of deciding about its behavior during runtime based on the dynamic behavior of its surroundings as the simulation is progressing. Hence, ABMs incorporate dynamically the effects of interactions with an agent's environment and neighbors, and conveniently account for the influence of random variations at agent level.
6. *Emergent behavior:* ABMs can generate complex emergent behaviors even with a few rules. They provide a powerful environment to describe emergence of behavior at one scale based on the events occurring at another scale.

Software agents can be equipped with different types of goals. These goals form the motivational component of agents and enable them to act proactively. Agent goals can be classified into the following taxonomy [26]:

1. *Performance goals* that the agents that are programmed to perform certain tasks must achieve
2. *Maintenance goals* which represent a state that agents want to maintain
3. *Achievement goals* which represent a desired state that the agent wants to reach
4. *Query (or test) goals* which represent an intention to obtain certain information

Additional goal types include combined goal types, such as "achieve then maintain" goals, which force an agent to reach a state and then to keep that state in later times [12]. Agents used for biomedical systems modeling usually possess only performance goals. Performance goals are a procedural type of goal and indicate that the goal of the agent is to execute certain well-defined actions. This is due to the fact that each agent in these systems is only aware of the functions it is entitled to perform, without knowing the higher level goal that the combination of agents, or the system, needs to pursue. This means that biological systems are modeled using individual goals for agents, rather than system goals for the entire system. Achievement of system goals is assessed by inspecting the emergent behavior, for example at tissue level.

Agents in medical systems are usually independent, intelligent, and mobile entities that perform specific tasks one expect from a cell (or its components). These agents are programmed to adopt any of the phenotypes a cell can posses as its state. These phenotypes may include motile, quiescent, proliferative, and apoptotic cells.

3 Agent-Based Modeling of Angiogenesis

The multi-layered ABM developed to simulate the process of angiogenesis can be used to investigate the effects of various factors on angiogenesis. The case studies

reported illustrate its use for identifying the role of pore size of a polymeric scaffold on implant vascularization. Most existing computational angiogenesis models consider solely the effect of soluble factors in the environment. A scaffold is desirable if the tissue volume to be replaced by engineered tissue is large.

3.1 Angiogenesis

Angiogenesis is the process of formation of new blood vessels from pre-existing vasculature [15]. It is a complex phenomenon that plays an important role in organ growth, healing and reproduction, and also in vascularizing tissue-engineered constructs. Abnormal angiogenesis can be initiated by diseases such as cancer and cardiovascular disease [9]. Consequently, angiogenesis has been the focus of a large number of experimental and theoretical studies that have provided information leading to successful outcomes such as recent anti-angiogenesis drugs.

Angiogenesis results in response to signalling of naturally occurring soluble factors, referred to as growth factors (GF), on endothelial cells (ECs), which are the cells lining inside of blood vessel capillaries. ECs become activated when they sense high concentrations of pro-angiogenic GFs. Through complex molecular and cellular mechanisms, some of the activated ECs differentiate into a special cell type, called tip cell. Tip cells start invading the surrounding tissue, following the direction of the GF gradient. Other factors such as insoluble ligand concentrations also play an important role in the process.

ECs behind the tip cell proliferate (cell division) and elongate (cell expansion) to fill the gap created between the motile tip cell and the host blood vessel. Combination of tip cell migration and EC proliferation and elongation results in formation of new capillaries. The newly formed capillaries connect to each other to make loops, a process called anastomosis, and once these loops are formed, blood flow starts in the new network. One of the features observed in sprouting capillaries is the persistence in branching and their migration direction: a new branch does not branch again immediately, and continues to move in the same direction for a time period known as its persistence.

Angiogenesis is a critical aspect of both tissue engineering and regenerative medicine as rebuilding tissues deeper than a few millimeters will not be successful without a healthy blood vessel network to supply nutrients, oxygen and other required factors [23]. Studying vascularization of synthetic scaffolds and implants can guide researchers in designing improved tissue engineering strategies that will lead to clinical success [7,20,10]. Computational models would enable researchers to study the effect of important factors while reducing the number of costly trial-and-error laboratory experiments.

3.2 Model Description

Our model [2] is implemented in Java and uses Repast (REcursive Porous Agent Simulation Toolkit), which is a Java-based open source agent modeling toolkit with features such as event scheduling procedures and visualization tools [22,18]. Repast includes packages of Java libraries that allow modelers to build simulation

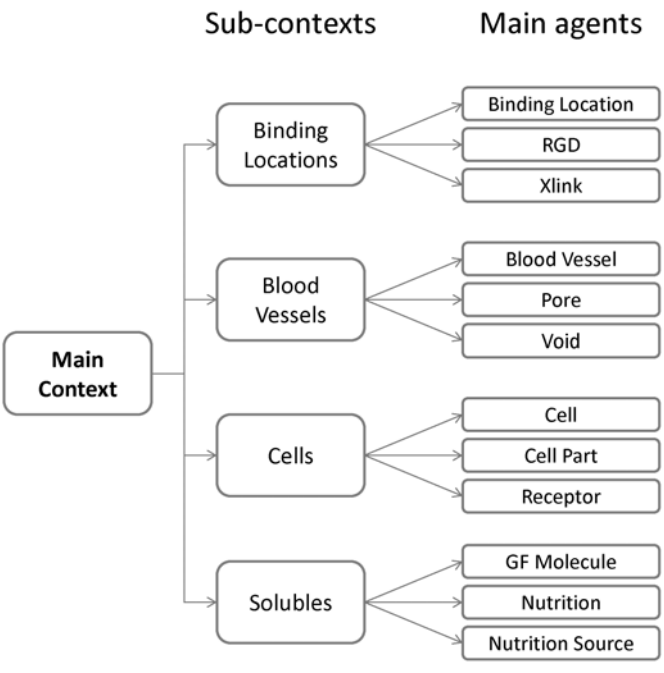

Fig. 1. Context structure of the model. The model includes four sub-contexts that contain different agent types. Sample agent types for each sub-context are shown.

environments, create agents in these environments, define relationships among the created agents, automatically collect data from simulation runs, and build user interfaces and displays in an easy fashion.

Repast is a widely used and complete Java-based platform [21], managed by the non-profit volunteer organization ROAD (Repast Organization for Architecture and Development). Repast has major benefits such as fast execution speed, inclusion of classes for network modeling, and ease of integration with the Eclipse IDE. Repast has the capability of developing multilevel models, utilizing the concept of context in designing the ABMs.

In Repast, different types of agents are organized by placing them in separate virtual buckets, referred to as *contexts* that enable hierarchical organization of the agents. Each context holds certain types of agents, and supports the control of relationships among the agents it holds through *projections*. A projection is a structure defined upon the agents of a context and its use enables the modeler to identify relationships among the agents in that context. The projections in Repast that are useful in modeling biomedical systems include network, grid, and continuous space projections.

The main context, which is the core object in Repast, holds all sub-contexts and their corresponding agents. Figure 1 shows the sub-contexts along with some of their main agent types. Our model structure includes four sub-contexts,

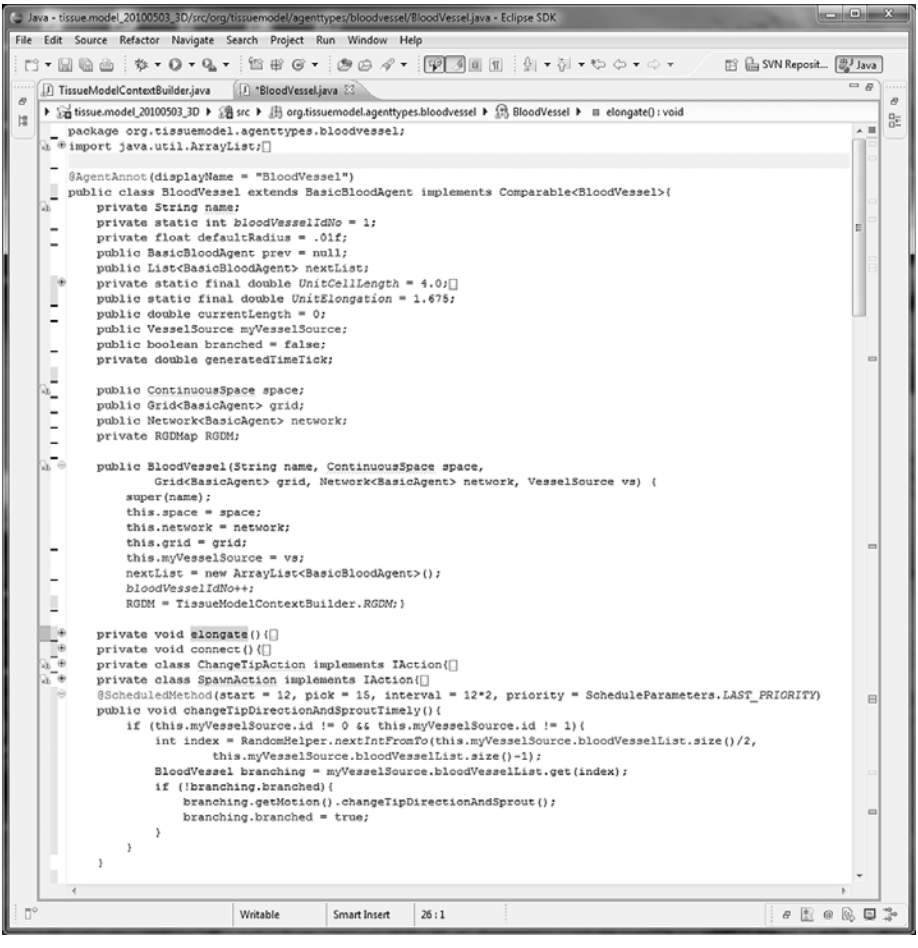

Fig. 2. Structure of BloodVessel class for creating EC agent objects. Only sample parts of the whole class are shown, including a number of instance variables, sample methods, and one scheduled method.

Binding Locations Context (for holding agents related to polymer molecules in the scaffold that ECs can attach to), Blood Vessel Context (that contains all the agents related to ECs), Cell Context (that includes all cell types other than blood vessel cells, and their parts and receptors), and Solubles Context (containing agents related to different soluble factors and the important soluble containers that hold the amount of each soluble at each grid point).

To illustrate the structure of agents, we have shown parts of an agent used in our model (Fig. 2). In this figure, selected parts of the BloodVessel class, which is the blueprint for creating BloodVessel agents, is depicted. This agent encapsulates a number of instance variables that determine the internal state of the agent. As an example, the currentLength variable, which is zero initially, holds

the length of each EC agent. When an EC elongates, the value of this parameter is updated, and controlled not to exceed the maximum allowable length, L_{max}. Each EC agent holds information of its previous (parent) EC agent, and a list of the next BloodVessel agents, in prev and nextList variables, respectively. The constructor, creates the BloodVessel object, and adds the space, network, and grid projections to the object. A number of methods define the behaviors the agent is capable of performing, such as connect (for connecting to other EC agents), and elongate (for increasing the length of the agent, EC growth). The method changeTipDirectionAndSproutTimely is an special method as it includes an annotation before its definition. This type of scheduled methods are used in Repast to perform tasks on a timely basis, as defined in the annotation.

The model developed includes a number of rules that govern the behavior of cells. The agents created interact with a virtual representation of the physical environment. Two types of agents have been considered to represent the two different cell types. However, in simulation runs we have only activated the EC agents to observe angiogenesis. Agents representing ECs mimic the actions that ECs perform during angiogenesis, which include elongation, proliferation, tip cell migration, and anastomosis, as governed by the rule base. Based on these rules, first the neighborhood of an agent is defined, and then the agents perceive their local environment and act based on their internal state and according to their embedded internal logic.

Figure 3 shows the flow chart of the time-driven processes, their connectivity, and the main model parameters [2]. EC agents conduct their actions either based on conditions in the environment (event-driven actions) or randomly based on time intervals (time-driven random actions). EC agents are proactive, as their internal state affects their behavior, and are non-adaptive because they have a determinate set of rules that are not being updated or altered during the simulation. In our model, an EC agent interacts directly only with its two neighboring EC agents in a blood vessel branch. A tip cell agent is an exception as it can move in the pore areas of the scaffold and has the ability to connect to another EC agent if they are in immediate neighborhood of each other.

Several layers in the model have been utilized to enable flexibility in handling the agents and representing their environment and neighborhoods [2]. A rectangular *grid* layer, defined using the respective projection in Repast, enables storing all the environmental information, including soluble concentrations and the details of porous scaffold structure at different grid points. Grid layer facilitates calculating gradients of solubles. Agents can find their neighbors by using this grid layer. Grid positions are discrete integer-valued coordinates. In contrast, agents are located and act on a continuous real-valued *space* layer, defined using the same projection of Repast. This combination of layers provides the tools to handle agents positioning and provide them with the ability to access information by querying their neighborhood. In addition to these two layers for handling agent positions, we use a *network* layer to keep track of the relationship between agents. Using this layer, we create a network of the agents that are connected to each other through parent and child relationship, and we save

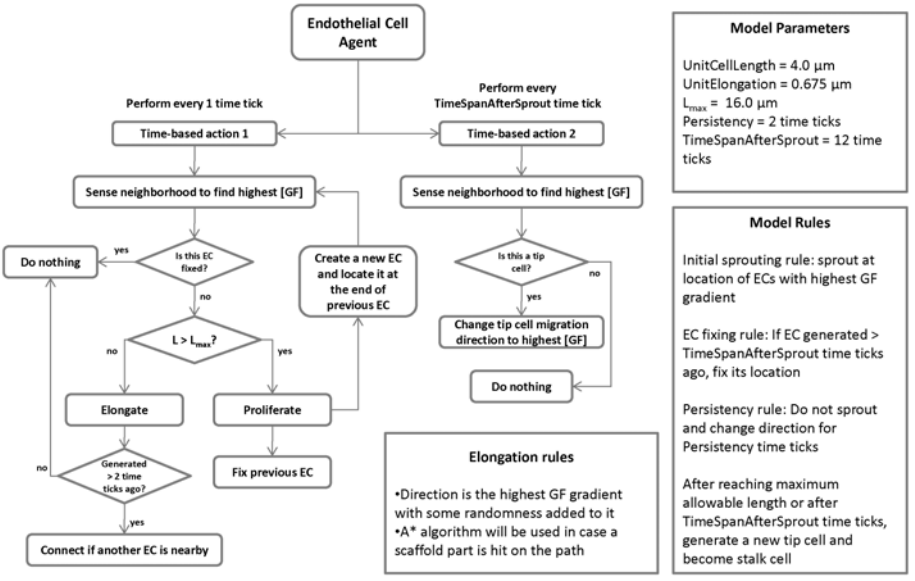

Fig. 3. Flowchart showing the main rules governing the behavior of tip cell and stalk cell ECs. Arrows indicate the connectivity of the rules.

these connection information to populate the number of ECs and sprouts in each of the branches at each time step in the simulation.

In our model, ECs are identified by two spherical nodes attached to each other with a connection in the network layer of the model. One of these nodes is designated to act for the EC itself, meaning that it is the agent assigned to represent that EC. The other node represents the parent EC of the current EC. The position of the EC node in the space layer is in fact the position of the front end of an EC. The rear end position of each EC is the location of its previous EC agent. As a result, the length of each EC would be the distance between its node and the node corresponding to its previous EC in the network layer. Using this abstraction method, we have been able to simply exhibit elongation of an EC by movement of its front node.

As a result of the time-based action 1 (Fig. 3), which happens at every time tick, tip cell node at each blood vessel sprout continues elongating until it reaches L_{max}. Once the length of an EC reaches L_{max}, proliferation occurs and another node is generated. The newly generated node becomes the leading node (tip cell) and continues elongation, while the previous node becomes fixed, which means it would not be able to move any further.

Fixed nodes can only become activated and create new leading nodes, causing branch formation and a new capillary vessel sprout, but they are not able to move themselves. Hence, blood vessels are able to change their direction either by proliferation and generation of new leading nodes or by branching. After a tip cell proliferates, the newly generated leading node has several options, including

elongation, proliferation, generation of new sprouts (branching), or connecting with other blood vessels based on its proximity to other vessel segments (anastomosis).

Proliferation has been modeled using the following algorithm: A tip cell EC proliferates either when it's length reaches L_{max}, or after a certain amount of time (TimeSpanAfterSprout) has passed from its generation. To create a new EC, first a new node is created at the location of current node, and then the current node retreats by half of the length of the proliferating EC. This mechanism leads to creation of two ECs of equal size. The previous node will then become fixed, and the newly generated node will become the tip cell and continue elongation in the same direction as the previous tip cell.

We have utilized a modified version of the A* search algorithm [14] for leading node pathfinding [2]. The leading node searches its local environment to find the location with the "lowest cost" among all the possible neighboring locations, while avoiding collisions with the scaffold. This cost is inversely proportional to the magnitude of GF concentration gradient. However, if the candidate location happens to be part of the scaffold, then a node cannot move to that location. We modified the algorithm so that the leading node does not compute its whole migration path from the beginning to the end *a priori*. The modified algorithm determines the node's path at each time step based on new information generated by the simulation. This enables the nodes to consider the effect of dynamic changes in environment (such as variations in GF concentration by production or consumption) as a result of presence of different cell types, and saves significant computation resources without reducing the efficiency of the algorithm as one tip cell will not be able to follow the whole path due to proliferation and the randomness embedded in it.

As a result of time-based action 2 (Fig. 3), at each time tick a certain number of stalk ECs (stalk cells are the cells that are not at the tip of a sprout, represented by fixed nodes in our model) will be chosen randomly and sprouting will happen at their location. The direction of the new sprouts will be in the direction of highest GF gradient at the location of the selected stalk ECs. These nodes will not be able to change their migration direction without making a new sprout. To consider the persistency of the sprouting branches, we have included a persistency rule in the rule base. According to this rule, a branch does not change its growth direction for a certain time, specified in the model with a persistency time parameter. In addition, a newly created branch cannot sprout to create another branch for the same persistency time.

The scaffold is modeled to have circular pores of equal size [2]. The diameter and shape of the pores can be varied to investigate angiogenesis in different pore sizes and structures. In the control case, all the space is pore region and hence available to the ECs. Vessel networks are limited to grow within the pores only and the scaffold locations are unavailable to EC agents. GF concentration (C_{GF}) profile in 3D runs (Equation 1) is a function of y and z directions. $Y_{max}/2$ and $Z_{max}/2$ are the maximum concentration locations, chosen to be the center of the scaffold (middle of the two existing blood vessels), and ε is a small positive

number to make the concentration at the initial locations greater than zero. A similar concentration profile have been used for 2D runs, which lack the z direction parameters. During simulation initialization, a random change of less than or equal to 0.01% of C_{GF} is added to the values of GF concentration to include stochastic behavior.

$$C_{GF}(y) = -(Y_{max}/2 - y)^2 + Y_{max}^2/4 - (Z_{max}/2 - z)^2 + Z_{max}^2/4 + \varepsilon \quad (1)$$

In 2D runs, we have simulated angiogenesis occurring on a 800×800 μm surface located in the first quadrant of the coordinate system. We initially place two main blood vessels, located at $y = 100$ and $y = 700$ μm. In 3D runs, due to memory limitations resulting from the presence of another dimension, scaffold size is reduced to 400×400×400 μm and four initial blood vessels are located at $y = 50$ and $z = 25$ μm, $y = 350$ and $z = 25$ μm, $y = 50$ and $z = 375$ μm, and $y = 350$ and $z = 375$ μm. These initial vessels represent the host vasculature, and the scaffold is implanted between them. The implanted scaffold is vascularized via sprouting angiogenesis from the surrounding host vasculature. EC agents possess an initial speed when they are first created which varies based on the local availability of scaffold attachment proteins (ligands), and based on the natural logarithm of the number of ligands in their immediate neighborhood.

The model presented in this paper and most ABMs of biomedical systems display a close relationship between the characteristics of the agents and biomedical phenomena. The technology should be considered in the development of ABMs for biomedical systems:

1. Recognition of the independent biomedical entities such as distinct cell types in the biomedical system that perform the most basic actions required to fulfill a goal, as agent classes. It is desirable to design the ABM with minimum number of agent classes.
2. Determination of the main behaviors of these entities that are essential for fulfilling their goals, and abstracting them into a number of functions. These functions must be designed in such a way to grant the agent independence in deciding its actions based only on what it perceives from its environment and its own state.
3. Design of the internal logic that will enable agents to perform their actions. This will be a rule-base for each agent type and it may include quantitative relations. In biomedical applications, this logic can include only simple if-then-else rules, or may incorporate sophisticated rules including sets of dynamic partial differential equations that model the gene-protein networks governing a cell's behavior.
4. Building an environment in which the agents function and selecting the environmental variables that are important for performing the actions of each agent type. The environment must be designed to include these variables, and the agents must have functions that enable them to survey their environment, and to identify their neighbors and their states. The environment in

biomedical applications usually includes tissue or extracellular matrix, and the variables can include the concentration of soluble or insoluble biological factors, or the matrix properties. It is possible to represent elements in the environment by dedicated agent types, but this approach requires more memory compared to using the built-in data structures of the programming environment used.

5. Definition of communication protocols between agents and with the environment.

The level of detail included in the model in each of these steps, will affect the accuracy and applicability of the model. However, there would be a compromise between the level of detail (and hence model accuracy) and the computational and memory costs. One challenge involved in translating from a biological process to an ABM is to decide the level of detail that results in useful simulation results, while it is feasible to run the model on available hardware.

3.3 Computational Challenges

One difficulty in developing ABMs for simulating biological systems is the computational limitations encountered, specially when the number of agents increase exponentially over time. In 3D the number of grid points may increase by 100 times depending on the size of third dimension, and hence memory becomes a limiting factor. One solution for this problem was proposed [2] by designing the 3D pores in an external software, such as 3ds Max [5], identifying the pore surfaces using a triangular mesh and providing the information of all the vertices and their normals. In this work, we overcame the memory limitation by using the 64-bit versions of Java jdk, Eclipse, and Java 3D on a 64 bit Windows 7 Professional edition that enables addressing as high as 192GB to memory. For 3D simulation runs, we used a work-station with 24GB RAM, on an Intel Pentium i7 processor that enabled a maximum grid size of $400\times400\times400$ μm.

In addition, we have made changes to the agent structure in our model to address the challenges regarding high memory cost of the model. As an example, we used the built in ValueLayer structures of Repast to represent scaffold and pore parts of the hydrogel, instead of using agents which required more memory. Using parallel computing techniques is a desirable way to overcome more extensive computational requirements encountered in modeling larger scale MAS systems. A single computer with multiple-core CPUs and large amounts of memory can also be considered for parallel computing. If the computational load distribution is performed based on the scaffold section in which the agents act, then the issue of agents that are crossing boundaries should be addressed since they are leaving one processing core and go into the control of another core. Da-Jun et al. have developed one sample platform for simulating a large multicellular ensemble of bacterial populations [11]. Their framework is aimed at running an ABM on a cluster-based hardware, using Message Passing Interface.

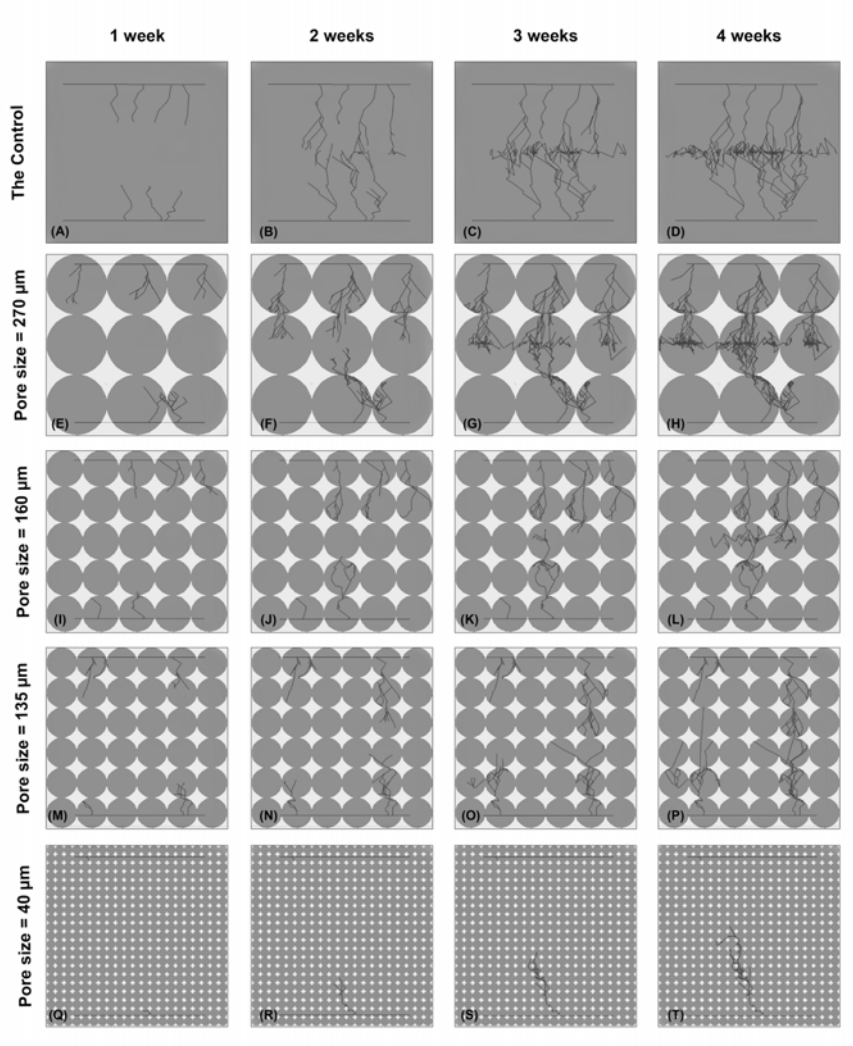

Fig. 4. Sample 2D simulation results for different cases after 1, 2, 3, and 4 weeks [3]. (A-D) The control, (E-H) Pore size = 270 μm, (I-L) Pore size = 160 μm, (M-P) Pore size = 135 μm, and (Q-T) Pore size = 40 μm.

4 Simulation Results

We have used the angiogenesis simulator developed to run case studies with different pore sizes in 2D. To adjust the speed of capillary growth, we first compared a number of simulation results with experimental results presented in [6], and we used this comparison to convert the simulation time ticks to real time in all subsequent case studies [3].

The parameters of interest are the time required for new capillaries to reach the center of scaffold and the number of sprouts after specific time (15 days in this case). Figure 4 depicts snapshots of 2D simulation runs for different cases at four consecutive times. A detailed description and evaluation of simulation results, along with comparison with published experimental data, can be found in [3]. Figure 5 shows a time series of the results of running the simulation in 3D for the control case, in which entire scaffold is available to ECs.

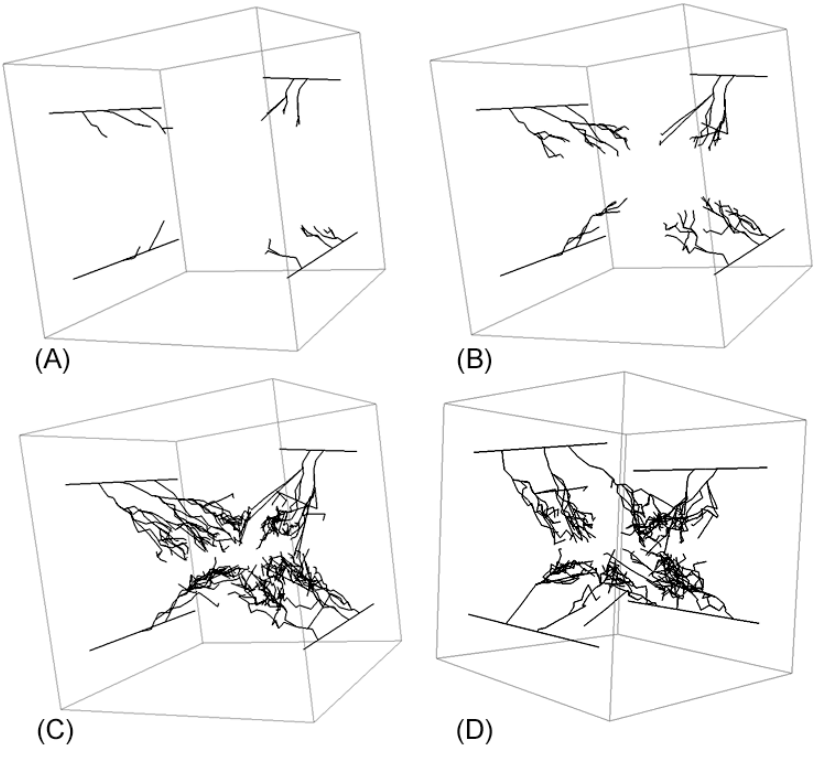

Fig. 5. Sample 3D simulation results for control case after 10 (A), 20 (B), and 30 (C, D) time steps. Picture (D) shows the same time step as picture (C) but from an opposite angle to illustrate the effect of 3D.

2D simulation results indicate formation of branching vascular networks from the host vessels. These branched blood vessels resemble 2D snapshots of blood vessel networks formed in in vivo angiogenesis in 2D assays [17]. 3D simulation results show how blood vessels fill the scaffold vascularizing it. The average time for blood vessels to reach the center of the scaffold is less than 10 days for the control case(Fig. 5).

The results of the simulation runs follow the same trends as experiments, indicating that the model is able to predict the basic behaviors expected from a developing blood vessel network. The rate of angiogenesis and the number of generated sprouts increase with increasing pore size, which supports the idea that larger pore sizes result in improved and faster angiogenesis in porous scaffolds. The simulation framework can also be used to predict the time for effective vascularization of the scaffold.

5 Conclusions

In this paper, we have shown how MAS can aid in addressing complex problems in biomedical engineering by offering computational tools required for the realistic simulation of biological systems comprised of a large number (approximately 100,000) of interacting biological cells. We have illustrated the advantages of using MAS concepts in simulating complex biomedical processes. The simulation framework developed is aimed at both understanding real world phenomena, and optimizing the way these phenomena take place. These models and simulations prove to be valuable specially when conducting experiments is costly and time consuming, which is the case with most biomedical experiments. The combination of computational and experimental results enables designing optimized biomaterials.

Acknowledgements. This paper is based upon work supported in part by the National Science Foundation (CBET-0731201) and the Veterans Administration.

References

1. An, G.: In silico experiments of existing and hypothetical cytokine-directed clinical trials using agent-based modeling. Crit. Care Med. 32, 2050–2060 (2004)
2. Artel, A.: Agent-based techniques in local, autonomous and adaptive decision-making. PhD dissertation: Illinois Institute of Technology (2010)
3. Artel, A., Mehdizadeh, H., Chiu, Y.-C., Brey, E.M., Cinar, A.: An Agent-Based Model for the Investigation of Neovascularization within Porous Scaffolds. Tissue Eng. Part A 17, 2133–2141 (2011)
4. Ausk, B.J., Gross, T.S., Srinivasan, S.: An agent based model for real-time signaling induced in osteocytic networks by mechanical stimuli. J. Biomech. 39 (2006)
5. Autodesk 3ds Max Products: 3D modeling, animation and rendering software, `http://usa.autodesk.com/3ds-max/`
6. Brey, E.M., McIntire, L.V., Johnston, C.M., Reece, G.P., Patrick, C.W.: Three-Dimensional, Quantitative Analysis of Desmin and Smooth Muscle Alpha Actin Expression During Angiogenesis. Ann. Biomed. Eng. 32, 1100–1107 (2004)
7. Brey, E.M., Uriel, S., Greisler, H.P., Patrick Jr., C.W., McIntire, L.V.: Therapeutic Neovascularization: Contributions from Bioengineering. Tissue Engineering 11, 567–584 (2005)
8. Brown, B.N., Price, I.M., Toapanta, F.R., Dealmeida, D.R., Wiley, C.A., Ross, T.M., Oury, T.D., Vodovotz, Y.: An agent-based model of inflammation and fibrosis following particulate exposure in the lung. Math. Biosci. 231, 186–196 (2011)

9. Carmeliet, P., Jain, R.K.: Angiogenesis in cancer and other diseases. Nature 14, 249–257 (2000)
10. Chiu, Y.C., Cheng, M.H., Uriel, S., Brey, E.M.: Materials for Engineering Vascularized Adipose Tissue. J. Tissue Viability 20, 37–48 (2011)
11. Da-Jun, T., Tang, F., Lee, T., Sarda, D., Krishnan, A., Goryachev, A.B.: Parallel Computing Platform for the Agent-Based Modeling of Multicellular Biological Systems. In: Liew, K.-M., Shen, H., See, S., Cai, W. (eds.) PDCAT 2004. LNCS, vol. 3320, pp. 5–8. Springer, Heidelberg (2004)
12. Dastani, M., van Riemsdijk, M.B., Winikoff, M.: Rich Goal Types in Agent Programming. In: Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011), pp. 405–412 (2011)
13. Grant, M.R., Mostov, K.E., Tlsty, T.D., Hunt, C.A.: Simulating properties of in vitro epithelial cell morphogenesis. PLoS Comput. Biol. 2, e129, Epub. (2006)
14. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 4, 100–107 (1968)
15. Levine, H.A., Nilsen-Hamilton, M.: Angiogenesis - A biochemical/mathematical perspective. In: Tutorials in Mathematical Biosciences III; Cell Cycle, Proliferation and Cancer, pp. 23–76 (2006)
16. Liu, G., Qutub, A.A., Vempati, P., Mac Gabhann, F., Popel, A.S.: Module-based multiscale simulation of angiogenesis in skeletal muscle. Theor. Biol. Med. Model. 8 (2011)
17. Muthukkaruppan, V.R., Kubai, L., Auerbach, R.: Tumor-induced neovascularization in the mouse eye. J. Natl. Cancer Inst. 69, 699–708 (1982)
18. North, M.J., Collier, N.T., Vos, J.R.: Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. ACM Transactions on Modeling and Computer Simulation 16, 1–25 (2006)
19. O'Neil, C.A., Sattenspiel, L.: Agent-based modeling of the spread of the 1918-1919 flu in three Canadian fur trading communities. Am. J. Hum. Biol. 22, 757–767 (2010)
20. Papavasiliou, G., Cheng, M.H., Brey, E.M.: Strategies for Vascularization of Polymer Scaffolds. J. Investig. Med. 58, 834–844 (2010)
21. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based Simulation Platforms. Review and Development Recommendations Simulation 82, 609–623 (2006)
22. Repast Home Page: Repast Organization for Architecture and Design, Chicago, IL, `http://repast.sourceforge.net/`
23. Rouwkema, J., Rivron, N.C., van Blitterswijk, C.A.: Vascularization in tissue engineering. Trends Biotechnol. 26, 434–441 (2008)
24. Thorne, B.C., Bailey, A.M., DeSimone, D.W., Peirce, S.M.: Agent-based modeling of multicell morphogenic processes during development. Birth Defects Res. C. Embryo. Today. 81, 344–353 (2007)
25. Thorne, B.C., Bailey, A.M., Peirce, S.M.: Combining experiments with multi-cell agent-based modeling to study biological tissue patterning. Brief. Bioinform. 8, 245–257 (2007)
26. van Riemsdijk, M.B., Dastani, M., Winikoff, M.: Goals in agent systems: A unifying framework. In: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), pp. 713–720 (2008)
27. Zhang, L., Chen, L.L., Deisboeck, T.S.: Multi-scale, multi-resolution brain cancer modeling. Math. Comput. Simul. 79, 2021–2035 (2009)