13

Agent-Based Modeling of Stem Cells

Mark d'Inverno
Goldsmiths, University of London

Paul Howells
University of Westminster

Sara Montagna
Università di Bologna

Ingo Roeder
University of Leipzig

Rob Saunders
University of Sydney

13.1 Introduction 389
Overview
13.2 The Biological Domain – HSC Biology 390
The Hematopoietic System • The Hematopoiesis Control Mechanisms
13.3 Stem Cell Modeling 392
Experimental Limitations • What a Model Can Be Useful For
13.4 Drawbacks of Existing Models and Why Agents . 394
13.5 Overview of Our Agent Modeling Framework 395
Framework Components • Interfaces between the Framework Components • Behavior of the Framework Components • Extending Our Agent Modeling Framework
13.6 Agentifying Existing Approaches 400
A Cellular Automata Approach to Modeling Stem Cells • Discussion about the Cellular Automata Approach • Re-formulation Using an Agent-Based Approach • Operation • Roeder-Loeffler Model of Self-Organization
13.7 From Agent Model to Simulation 407
MASON • Implementation of CELL in MASON • Implementation of Roeder-Loeffler Model in MASON
13.8 Discussion 412
Concluding Remarks
Acknowledgments 415
References 415

13.1 Introduction

The multi-agent systems approach has become recognized as a useful approach for modeling and simulating biological complex systems. In this chapter we provide an example of such an approach, which concerns the modeling and simulation of the Hematopoietic Stem Cell (HSC) system in adults. We are specifically interested in how local cell interactions give rise to well understood properties of systems of stem cells, such as the ability to maintain their own population and to maintain a population of fully differentiated functional cells. There is a need to establish key cell mechanisms that can produce self-regulating behavior of HSC systems using different theoretical techniques. It is our belief that modeling the behavior of HSCs in the adult human body as an agent-based system is the most appropriate way of understanding these mechanisms and the consequent process of self-organization.

In recent years there has been a growing debate about how stem cells behave in the human body; whether the fate of stem cells is pre-determined or stochastic, and whether the fate of cells relies on their internal state, or on extra-cellular micro-environmental factors. However, current experimental limitations mean that stem cells cannot be tracked in the adult human body. There is no way of "observing" micro-level behavior. Models and simulations have a crucial role therefore in explaining the relationship of micro-behavior to macro-behavior and it now seems that the importance of computational modeling and simulation for understanding stem cells is beginning to be realized in many wet-labs. There have been several attempts to build formal models of these theories, so that predictions can be made about how and why stem cells behave, both individually or collectively. In this chapter we propose an agent based model which describes at the same time the intracellular behavior of the cell (i.e., intra-cellular networks) and the cellular level where all the systemic interactions are developed. This enables us to build a multi-level model.

13.1.1 Overview

In Section 13.2 we provide an overview of the background biology for this investigation. We then discuss the current experimental limitations and how these motivate the significance for using formal, computational models. We then describe the main approaches of formalizing conceptual models of stem cells, which can be classified as single cell vs population models, stochastic vs deterministic models. By discussing these limitations we are able to clearly promote the advantages of the agent approach. Not only in understanding how cell-cell interactions give rise to overall system behavior but also because they provide an appropriate level of abstraction for collaborating with biologists.

Then we describe out agent modeling framework and how we have used it to take existing formal modeling approaches and "agentify" them to address certain weaknesses in the model. Finally, we will provide details of the simulation and discuss our results to date and plans for future work.

This chapter is intended for an interdisciplinary audience and so we include introductory material about the background biology as well as about why the agent approach is most suitable in this domain.

13.2 The Biological Domain – HSC Biology

Due to the natural life-cycle of cells, a tissue has to be self-renewing: in order to guarantee a continuous replacement of cells that die, naturally or after injury, a cellular population must contain cells that are able to proliferate, generating a mixture of progeny. The population of daughter cells then includes cells that remain undifferentiated, i.e., self-renew the identity of their parent, and cells that differentiate, i.e., change their properties. Cells with the potential for both self-renewal and differentiation are called *stem cells*.

Even though no commonly agreed phenotypic or molecular definition of a stem cell exists, there is a general consent on their functional capabilities. A stem cell is an undifferentiated cell that is able to (i) proliferate, (ii) self-renew, i.e., able to go through numerous cycles of cell division while maintaining the undifferentiated state, (iii) differentiate, i.e., able to produce a progeny of distinct cell types, (iv) recover the tissue after injury or disease. Those capabilities and their importance for a functional definition of tissue stem cells has extensively been discussed by [Potten and Loeffler, 1990] and [Loeffler and Roeder, 2002].

It is still not clear whether stem cell types for different tissues are strictly committed to these tissues or whether the can flexibly adopt features of other tissues under certain

circumstances. However, there is increasing experimental evidence for the flexibility and the reversibility of stem cell function and phenotype [Blau and Blakely, 1999; Quesenberry et al., 2001].

Furthermore, it is evident (see citation given above) that the emergent behavior of the stem cell system depends on multiple factors, such as their own actual cellular state, their interaction with other cells or environmental cues.

13.2.1 The Hematopoietic System

Hematopoietis is the entire process of production and maintenance of all types of blood cells which exhibit very different functions, such as transport of oxygen, production of antibodies to fight infection and blood clotting. The life-span of blood cells, however, is limited so that they must be continuously produced throughout the life of the animal. The hematopoietic stem cells (HSCs), which can be found mostly in the bone marrow, are responsible for the constant replacement of blood cells lost to normal turnover processes as well as to illness or trauma.

HSCs are able to generate every lineage found in the hematopoietic system through a successive series of intermediate progenitors. An exhaustive representation of the HSCs lineage tree is published in the *Kyoto Encyclopedia of Genes and Genomes*, see [Kanehisa and Goto, 2000]. Although the process of lineage specification is continuous, it is possible to identify some main phases, the first of which includes common lymphoid progenitors (CLPs), which can generate only B, T, NK cells, and common myeloid progenitors (CMPs) which can generate only red cells, platelets, granulocytes, and monocytes. In the following phases there are more mature progenitors that are further restricted in the number and type of lineages that they can generate. The last phase is the terminally differentiated cell that cannot divide and that undergoes apoptosis, i.e., programmed cell death, after a period of time ranging from hours to years.

In this way, during homoeostasis, a proportion of stem cells is expected to balance the fundamental processes of self-renewal, differentiation, apoptosis and quiescence to maintain (i) a constant flow of short-lived progenitors that can generate enough cells to replace those that are constantly lost during normal turnover and (ii) a constant number of primitive cells to sustain hematopoiesis. Under homoeostatic conditions this results in a number of tissue stem cells, as well as blood cells, that fluctate around a relatively constant average value. During times of physiologic stress, the entire hematopoietic system is then able to provide the mature cells required to fight infections or to replenish cells lost during a hemorrhage. When the stress is resolved the number of cells returns again to normal levels.

A variety of homoeostatic mechanisms allow blood cell production to maintain homeostasis or to respond quickly to stress such as bleeding or infection; the next section provides a brief account of some of these mechanisms.

13.2.2 The Hematopoiesis Control Mechanisms

The process of hematopoiesis involves a complex interplay between the intrinsic genetic processes of HSCs, progenitors and mature blood cells, that can be either deterministic or stochastic, and their microenvironment dynamics, responsible for cell to cell interactions. As a consequence of these internal and external processes, these cells remain quiescent, differentiate, self-renew, or undergo apoptosis.

Cell intrinsic regulatory processes are cell autonomous and are determined by the cell's developmental state, which translates into specific levels of genes and protein expression. With these processes we refer to intracellular pathways and gene regulatory networks. De-

pending on the number of molecules involved they appear to be deterministic or stochastic. The intracellular events are in fact determined by the interactions between molecules: as the number of reacting molecules increases the probability of their interaction increases until a threshold over which the process can be considered deterministic.

The role of the microenvironmental dynamics is also crucial. The specific microenvironment of stem cells has been historically called "stem-cell niche" [Schofield, 1978]. A niche is composed of a dynamically changing chemical environment, containing a range of different molecules. Cells maintain this environment by secreting and absorbing molecules, in order to send and receive signals. The information passed in these signals influences the autonomous behavior of the cells, e.g., *growth factors* stimulate cells to divide and differentiate to produce more terminally differentiated cells.

All of the genetic and environmental mechanisms that govern blood cells production operate by affecting the intracellular level dynamics, resulting in different cell actions and behaviors. Under normal conditions, the majority of HSCs and many progenitors are quiescent. In the event of a physiologic stress quiescent progenitors and HSCs are stimulated by a variety of growth factors to proliferate and differentiate into mature white cells, red cells, and platelets. When the bleeding or infection ceases and the demand for blood cells returns to normal, the antiapoptotic and proliferative stimuli decreases and the kinetics of hematopoiesis return to baseline levels [Wichmann and Loeffler, 1985; Smith, 2003; Attar and Scadden, 2004; Wilson and Trumpp, 2006].

13.3 Stem Cell Modeling

To understand the behavior of hematopoietic systems, it is not enough to study the single cell behavior, but we need a complete view of the system, with the overall mechanisms that govern the internal dynamics of the cells and their communications through the microenvironment.

13.3.1 Experimental Limitations

We believe that recent experimental evidence makes it clear that it is increasingly necessary to use formal, computational models to investigate the nature of stem cell systems rather than stem cells in isolation. This is for several key reasons. First, adult stem cells cannot be easily isolated, indeed it may be that it is only by looking at their behavior in a system, not isolated, can we tell what kind of cell we were originally looking at. Second, any attempt to determine the properties of stem cells requires a functional test, which itself feeds back on the cells and changes their properties [Potten and Loeffler, 1990; Loeffler and Roeder, 2002]. Third, even if we were able to track the behavior of a cell in the body, it would only tell us about one of the possible behaviors of the original cell; it tells us nothing about the potentially considerably larger array of behaviors that may have been possible if the environment and the chance elements had been different. Fourth, by removing a cell from its original and natural habitat the new environmental conditions will influence future behavior and lead to misleading results [Theise, 2005]. Fifth, it is the totality of the stem cells as a system in the human body that is important. A key quality of the system is its ability to maintain exactly the right production of cells in all manner of different situations.

In response, therefore, we have developed a formal model that reflects many of the key experimental and recent theoretical developments in stem cell research. Using techniques from multi-agent systems, we are currently building a complex adaptive system to simulate stem cell systems in order to provide a testbed from which to be able to investigate key

properties of them in general and to formulate new experiments to identify the underlying physiological mechanisms of tissue maintenance and repair.

We next outline some of the techniques that have been employed up till now.

13.3.2 What a Model Can Be Useful For

The mathematical modeling, conceptualization and simulation of stem cell behavior is beginning to receive a substantial amount of interest from an increasing number of researchers. As has been pointed out by others, predictive models of stem cell systems could provide important new understandings of the self-regulating mechanisms that result in the global properties of stem cells.

There has been a growing debate about how stem cells behave in the human body; whether the fate of stem cells is pre-determined [Nicola and Johnson, 1982] or stochastic [Ogawa, 1999; Thornley et al., 2003], and whether the fate of cells relies on their internal state [Novak and Stewart, 1991], or on extra-cellular micro-environmental factors [Trentin, 1970]. There have been several attempts to build formal models of these theories, so that predictions can be made about how and why stem cells behave as they do, either individually or collectively. Reviews of these formal approaches can be found in recent publications [Sowmya and Zandstra, 2003; Loeffler and Roeder, 2004; Roeder, 2006] and we do not propose to review these models in this chapter.

It is worth noting, however, that the first model we know of was published in 1964 ([Till et al., 1964]) and that there has been surprisingly little work in this field until the last couple of years. There has been a noticeable climate change in this respect, and there is now a growing awareness of the need to use mathematical modeling and computer simulation to understand the processes and behaviors of stem cells in the body.

We summarize what we see are the key reasons for the systematic development of models and simulations to consider hypotheses about the nature and behavior of stem cells.

1. The size and complexity of stem cell systems mean that without simulation, it is practically impossible to consider the whole system. Simulations provide an important tool for understanding the global behavior of complex systems.
2. Clearly any model, and resulting simulation, of stem cells will necessarily incur massive simplifications and abstractions about the machinations of the human body. It is our belief, however, that theoretical simplifications are often key to understanding fundamental properties of natural systems.
3. It is the *potential* of cells to behave in lots of different ways which makes them more or less stem like. It may be that stem cell is a notion rather than a concrete entity and refers to the wide-ranging set of potential behaviors that a cell might have that are influenced by internal, environmental, and stochastic processes. Simulations provide a way of determining which behaviors are essential to stem cells and which are incidental in systems that have been studied in the laboratory.
4. When you consider experimental evidence you have seen only one behavior. This behavior may have been one of many, and it is the potential for cells to behave in certain ways that might be key to defining them. Modeling and simulation is a much more effective device for understanding "behavioral potential" than looking at completed chains of events in the lab.
5. Simulation is cheap.

13.4 Drawbacks of Existing Models and Why Agents

As we have pointed out in the previous sections, we believe that studying the behavior of hematopoietic system needs the comprehension of both intracellular processes and interacting events among cells that are mediated by the cells dynamic microenvironment. In order to understand the evolution of the cellular population, we should look at the system at least, at two levels: the intracellular level and the extracellular one. We therefore require a *multi-level model*, which is the one that best fits with the biological problem we are proposing [Uhrmacher et al., 2005].

Most of the existing approaches to modeling and simulating biological systems can capture details at only one level. The most common approach concerns differential equations, in different forms: ordinary, partially or stochastic differential equations (ODE, PDE, SDE). A huge amount of literature, especially in the systems biology discipline, describes ODE models of metabolic pathways, signalling pathways and gene regulatory networks [Conrad and Tyson, 2006]. The results obtained at this level with this approach gave good insights on biological systems behavior. When the system shows non-deterministic dynamics, then the introduction of stochastic approaches such as chemical master equations (CME), stochastic simulation algorithms (SSA), tau leaping, chemical Langevin equations and so on [Gillespie, 2008] allow the study of stochastic events. Alternatively a lot of other different approaches have been applied: random boolean networks [Kauffman, 1993], Bayesian networks and more recently, computational models such as different process algebras (an interesting example is the work that has been done with stochastic *pi*-calculus [Regev et al., 2001] and its extensions [Dematte et al., 2008]), or petri-nets [Talcott, 2008; Heiner et al., 2008]. An excellent review is given by [De Jong, 2002].

All of these approaches have their peculiarities, advantages and drawbacks, which make them useful under certain circumstances, and for specific biological systems. But they share a property: they hardly can be used for a multi-level model.

In this context we can motivate the agent approach by noting the following,

1. An agent-based approach provides more flexibility than other more limited approaches and so delivers greater potential for modeling more sophisticated, globally emergent, behavior both on the individual cell and on the cell population level.
2. We can explicitly represent an environment.
3. An agent-based approach provides more biological plausibility than existing approaches such as cellular automata and other mathematical approaches. One of the main reasons that biological plausibility is important is to attract biologists to use and work with any models and simulations that are created.
4. We can build multi-layer models that capture micro- and macro-level features of the biological system.
5. Stem cells are a prime example of a self-organizing system where individual cells react to their local physical, chemical and biological environment. The system should therefore be most suitably modeled as a system of interacting reactive agents, where the reaction at the micro level gives rise to the emergent behavior at the system level.
6. Even though we are simulating cells and environment, the Brooksian idea of an agent being something which is both situated and embodied [Brooks, 1991] is a fundamental driving force of our use of agents as the appropriate modeling paradigm. Cells modeled as agents have a physical, chemical and biological pres-

ence and are situated in a physical, chemical and biological environment in which they react. The way in which they react will then influence the way other cells react in the future and so on.

7. Agent-based models and simulation provide us with an easy facility to perform "what if" experiments. By changing the behavior of individual cells or environmental conditions we can perform experiments to see how system behaviors might be affected.
8. Agent systems lend themselves to elegant visualizations which have meaning to and impact on, the stem cell biologist.

In addition, by situating our simulation work in a wider formal framework, the SMART agent modeling framework, based on years of previous investigation [d'Inverno and Luck, 2004; d'Inverno et al., 2004] we can compare and evaluate different models. We believe that this is necessary for this new field to develop in a systematic manner. Moreover, the formal framework allows us to "agentify" existing models, making it very clear what the relationship between the existing version and the agent version is. We shall see this later in the chapter. Finally, by building a formal model of stem cell behavior in an agent framework and using a specification language from software engineering, there are techniques to ensure that the simulation correctly implements the model.

The application of multi-agent based simulation for the analysis of biological systems is quite recent, but already some work has been done in this area, for example, [Merelli et al., 2007; Montagna et al., 2007, 2008].

13.5 Overview of Our Agent Modeling Framework

We begin the overview of our stem cell agent modeling framework by first describing and explaining the purpose of the the main components of the framework: *cell agents*, the *environment* and the *simulation engine*. Next we describe the interfaces between the various components and finally we describe the behavior of the components.

A diagrammatic overview of the framework is given in Figure 13.1. The role of each of these framework components is as follows:

Cell Agents: these are used to model the individual cells of the system being modeled.

Environment: represents the biological, chemical and physical environment that the cells exist in.

Simulation Engine: the purpose of this is to "drive" a simulation of a stem cell system. It does this by updating the environment in response to requests and signals from the individual cell agents.

An important point to note is that the introduction of an environment that includes chemical and topological information allows us to reduce the need to rely on stochastic processes, i.e., probability functions. Moreover, it allows us to actually model the physical movement of cells which clearly provides us with a more sophisticated and detailed model of self-organization in stem cell systems. We will see examples of the advantage of modeling physical aspects in agent-based models later in this chapter as we agentify existing formal models.

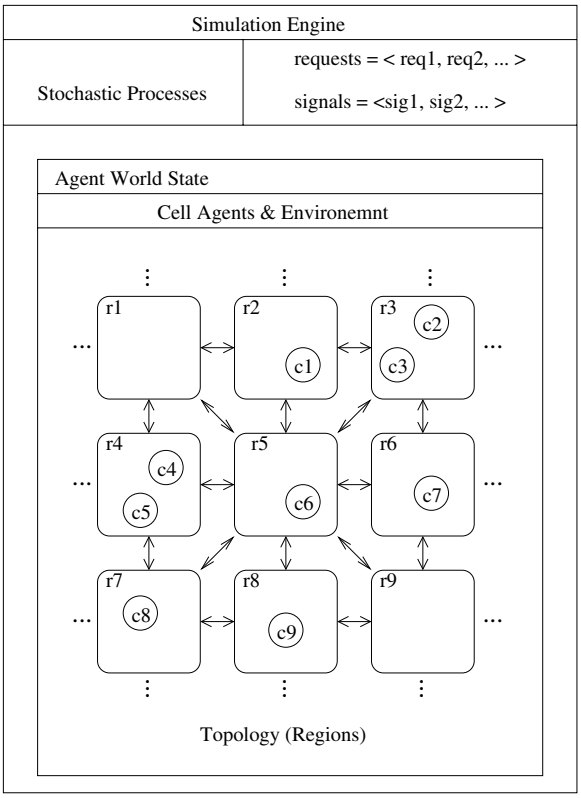

FIGURE 13.1 The Agent Framework for Stem Cell modeling. Where the ri are regions and ci are cell agents.

13.5.1 Framework Components

In this section we describe the *state information* that is contained in each of the three framework components: *cell agents*, the *environment* and the *simulation engine*.

These components can be viewed as forming subsystems: the *Agent World State* that comprises the cell agents and the environment; and the *Stochastic World State* which adds the stochastic processes to *Agent World State*.

The *Agent World State* is defined as a combination of the cell agents, the environment and the locations of the cell agents in the environment:

- The agent world state is composed of two components the cell agents and the environment.
- A cell is modeled as an agent and consequently a cell agent's state consists of biological and operational information.
- In addition, a cell agent has a number of physical properties, for example, location, orientation and possibly velocity and momentum as a result of being in the environment.
- The environment is composed of the following sub-components chemical and physical laws and a topology of regions.
- The individual regions of a topology have attributes that can be *perceived* by the cell agents.

The request mechanism, described later in this chapter, allows the modeling of cells to focus on the logic of their operation and leaves questions of physical simulation, e.g., collision detection, to the simulation engine.

Cell Agent

A cell agent is used to model a biological cell, but it is also necessary that it include sufficient attributes to be able to function as an autonomous agent. So in our framework a cell agent's state is composed of four main components:

- The internal state of the cell that represents its biological state.
- The cell agent's percepts, that is the information that a cell can perceive from its environment.
- The cell agent's plan, these are the actions that the cell has decided to attempt.
- A cell may also have a globally defined stochastic component that needs to be added purely for modeling purposes.

The second and third components are concerned with the operational aspects of a cell agent, that is the *perceive*, *plan* and *act* agent behavioral cycle. At this level of model description the precise nature of the plans developed by the agents is left open. Planning may be implemented using approaches like those typically used in Belief-Desire-Intention (BDI) agents or, as in the case of the work described later, using a reactive planner. The last component is not consistent with a true agent approach because it introduces a global level of control but is a necessary component when agentifying existing systems, e.g., the Roeder/Loeffler model [Roeder and Loeffler, 2002].

Environment

We define the notion of an environment using the general notion of a *topology*, which is defined as a set of adjacent *regions*. Cells are located within a region, and each region may contain from zero up to a predetermined maximum number of cells. Further each region has associated with it chemical concentrations, physical forces and so on. So for example, in a particular model a region may contain at most one cell, thus it may be either empty or contain a stem, progenitor, determined or stromal cell.

So the environment is composed of the following components:

- A *topology*, defined in terms of a set of *regions* and an *adjacency* relationship between them. These properties are fixed for a particular model.
- An individual region's *static attributes*, e.g., how many cells it can contain. These properties are fixed for a region in a particular model.
- An individual region's *dynamic attributes*, e.g., chemical concentrations and the cells it contains. These properties can vary for a region in a particular model.
- A mapping which represents the *location* of the cells within the environment, i.e., the region it is currently in.
- Chemical and physical laws, e.g., chemical diffusion rates. These properties are fixed across all models.

Simulation Engine

The simulation engine is able to see the state of all components in the system being modeled, in particular the cell agents and the environment, thus it is able to extract any

system state information it requires from these components. In addition, the simulation engine also has the following internal state information:

- The signals and requests it has received from the cell agents.
- The state of any stochastic processes that it is using to provide stochastic inputs to the cell agents.

The simulation engine used in the following implementations works in a discrete step-wise manner, but the high level of description used at the modeling phase permits for other types of simulation engines, including discrete event simulations with asynchronous execution. For more information on different types of simulation engines see the chapter by Theodoropoulos et al. in this book [Theodoropoulos et al., 2008].

13.5.2 Interfaces between the Framework Components

We now describe the *interfaces* between the cell agents, environment and simulation engine.

Cell Agent and Environment

Each cell agent *perceives* information from its environment, that is it inputs the "state" of its current environment. Its current environment is defined as its current location which will consist of at least its current region and possibly the adjacent regions. This information is input in the form of *percepts*. In general these will be chemical, physical and biological information. For example, a cell agent can perceive its current location (region) as well as the chemical concentration in that location.

Cell Agent and Simulation Engine

The cell agents *issue requests* to the simulation engine to perform a "global" event and *send signals* to notify the "environment" that it has performed a unilateral action, for example secrete chemicals.

Examples of cell agent *requests* would be:

- to move to another location; or
- a parent cell to divide and be replaced by two daughter cells; or
- a cell to die and be removed from the environment.

In practice this is achieved by a cell agent sending its request to the simulation engine, which then appends the request to its list of pending requests. Some of a cell agent's planned (internal) actions do not require the simulation engine to do anything, and hence no request is generated.

Depending on the model under consideration a cell agent would be required to input stochastically generated information from the simulation engine.

Simulation Engine and Environment

The simulation engine simply retrieves state information from the environment and updates the state of the environment as a result of processing the cell agents' requests and signals. The simulation of physical processes, such as chemical diffusion, are handled by the simulation engine, rather than being embedded in the environment, allowing freedom of implementation.

13.5.3 Behavior of the Framework Components

In this section we discuss the behavior of the cell agents and the simulation engine. The environment is a passive repository of information and hence does not have an active behavior as such.

Cell Agent Behavior

In agent-based computing the operation of an agent is given by the *perceive*, *plan* and *act* cycle. In particular, what an agent perceives depends on its internal state and that of the local environment; what it plans to do depends on its perceptions and internal state; and what action it actually does depends on what it plans to do and the state of the environment.

The above general agent behavior cycle is refined for a cell agent as follows:

- The *perceive* action is a passive engagement with the environment, that simply involves inputting *percepts*. In general we aim to model chemical, physical and biological perceptual abilities. For example, a cell agent can perceive the chemical concentration in its current location (region).
- A plan to perform one of its possible actions is formulated after the cell agent has perceived its environment, and then which action it decides to add to its plan is determined by these percepts and its internal state. The types of actions that a cell agent can choose to do are remain in its current location, move to a new location, divide, secrete chemicals or ultimately die. In the framework a cell agent's plan consists of several types (see below) of actions: *internal actions* and their corresponding *requests* and *signals* to the simulation engine. The requests and signals are determined by the planned internal action.
- The agent *acts* by performing the first action in its plan, the remaining plan is then just what is left after this action has been removed. The generalized cell agent's *action* behavior can further be divided into the following sub-actions of *send requests*, *send signals* and *do action*. Where these represent the cell agent sending a request and signal respectively to the simulation engine, and the performance of the cell's planned internal action respectively.

Note that in some types of system the cell agent's behavior cycle would include the inputting of stochastically generated information calculated globally by the simulation engine, in a true agent model this would not be necessary. This is not part of the cell agent's perceptions but part of the stochastic modeling technique for certain models. When the cell agent's behavior cycle is implemented we assume that it would be *atomic*, in the sense that after an agent has perceived its environment the information that it has perceived, i.e., percepts, remain true throughout the remainder of its behavior cycle.

In our framework we have identified three types of "actions" that can form a cell agent's plan:

Internal actions: these are actions that a cell agent can perform independently of its environment, these actions only effect its internal state and do not require the participation of its environment.

Signals: these represent a cell agent *signaling* to its environment. These signals do not require the environment's participation in the sense of a joint action, but

only that the environment records the effect of the signal having occurred*. For example, when a cell decides to secrete a chemical, this would be modeled by a signal.

External requests: these are requests that a cell agent makes to its environment. They are associated with the cell's (internal) actions and are used when a cell's chosen action requires the participation of the environment for the complete effect of the action to occur. If the environment is unwilling or unable to collaborate with the action, i.e., grant the request, then it can not be performed, in these circumstances the agent would continue to *wait* for the request to be granted. This allows us to model an action that effects the state of the cell and environment, for example, cell division or death.

Simulation Engine Behavior

In the current model the main purpose of the simulation engine is to "run" the simulation, it does this by processing the action requests and signals generated by the cell agents. For example, each request is processed as appropriate by either:

- moving a cell to a new location (region); or
- deleting a dead cell from the system; or
- deleting the parent cell from the system and creating and adding two new daughter cells; or
- if the request can not be granted taking no action.

In general the simulation engine may have to deal with any conflicts that might arise due to incompatible requests.

The simulation engine is also responsible for:

- Updating the state of the environment, for example, a regions chemical concentrations when a cell agent has secreted a chemical.
- It is also responsible for generating stochastic information using stochastic processes, e.g., probability functions, when this is required by the system under consideration.

13.5.4 Extending Our Agent Modeling Framework

We are not fixed on this model. It is important that as we incorporate different models, perspectives and experimental findings into our approach that the model can develop. One of the ways we validate and grow this model is to apply this agent-based modeling approach to "agentify" other formal models. We discuss two case studies in the next section.

13.6 Agentifying Existing Approaches

To demonstrate the validity and applicability of our agent approach we consider two recent models in more detail.

*This is similar to Roscoe's [Roscoe, 1997] view of the process termination event *tick*, in the process algebra CSP.

13.6.1 A Cellular Automata Approach to Modeling Stem Cells

In recent work, Agur et al. [Agur et al., 2002] built a cellular automata (CA) model to show how the number of stem cells in the bone marrow could be maintained and how they could produce a continuous output of determined cells. The bone marrow is considered to be a stem cell *niche* where most biologists believe that the human body's supply of hematopoietic stem cells are situated and maintained.

This work is important because it is one of the few examples where a mathematical model has been used to investigate properties of stem cells that might be required to enable the maintenance of the system's homeostasis. The model demonstrates a possible mechanism that allows a niche to maintain a reasonably fixed number of stem cells, produce supply of mature (determined) cells, and to be capable of returning to this state even after very large perturbations that might occur through injury or disease. The behavior of a cell is determined (equally differentiated) by both internal (intrinsic) factors, e.g., a local counter, and external (extrinsic) factors, e.g., the prevalence of stem cells nearby, as stated by the authors as follows.

1. Cell behavior is determined by the number of its stem cell neighbors. This assumption is aimed at simply describing the fact that cytokines, secreted by cells into the micro-environment are capable of activating quiescent stem cells into proliferation and determination.
2. Each cell has internal counters that determine stem cell proliferation and stem cell transition into determination as well as the transit time of a differentiated cell before migrating to the peripheral blood.

In order to demonstrate this model we will provide a small part of an existing specification of this system. The specification is written in the Z specification language [Spivey, 1992]. The full specification can be found elsewhere [d'Inverno and Saunders, 2006] but we provide a taste since the formal background to our agent-based approach is a key part of this work. In the cellular automata model, the stem cell niche is modeled as a connected, locally finite, undirected graph.

$$graph : Node \leftrightarrow Node$$
$$neighbors : Node \rightarrow (\mathbb{P}\, Node)$$

Any *Node* is either empty, or it is occupied by either a stem cell or a determined cell.

$$TypeAg ::= EmptyAg \mid StemAg \mid DeterminedAg$$

The state of any node is given by the node location, the state, and an internal clock.

NodeStateAg
$$node : Node$$
$$type : TypeAg$$
$$counter : \mathbb{N}$$

The set of all such nodes is then given below, and defines the system state. We also define a function that returns the neighboring node states for any given node state.

SystemStateAg
$$nodes : \mathbb{P}\, NodeStateAg$$
$$neighborsAg : NodeStateAg \rightarrow (\mathbb{P}\, NodeStateAg)$$

There are three constant values, we will call them *LeaveNicheAg*, *CyclingPhaseAg*, and *NeighborEmptyAg* in our specification, that are used to reflect experimental observation. *LeaveNicheAg* represents the time taken for a determined cell to leave the niche. *CyclingPhaseAg* represents the cycling phase of a stem cell; a certain number of ticks of the counter are needed before the cell is ready to consider dividing. Finally, *NeighborEmptyAg* represents the amount of time it takes for an empty space that is continuously neighbored by a stem cell, to be populated by a descendent from the neighboring stem cell.

$$LeaveNicheAg, CyclingPhaseAg, NeighborEmptyAg : \mathbb{N}$$

We now specify how the system changes over time. Whenever there is a change of state in the system, we identify the node that we are considering as *node*. All locations are updated simultaneously.

The rules of this model, which determine what happens at a node based on internal and external factors are described and specified below.

1. Determined cell nodes
   (a) If the internal counter of a node representing a determined cell has reached *LeaveNicheAg* then the cell leaves the niche; the internal counter of the node is reset to 0, and the new state at the node becomes empty.
   In the schema below, the variable *node* represents the node determining its next state, and the variable *newnode* is that new state. We show one operation only, the rest are very similar.

   DeterminedLeaveNicheAg
   $\Delta SystemStateAg$

   $node.type = DeterminedAg$
   $node.counter = LeaveNicheAg$
   $newnode.type = EmptyAg$
   $newnode.counter = 0$

   (b) If the internal counter has not yet reached *LeaveNicheAg* then the internal counter is incremented.
2. Stem cells nodes
   (a) If the internal counter of a node representing a stem cell has reached the constant *CyclingPhaseAg*, and all of the nodes neighbors are stem cells, then the state of the node becomes a determined cell and the internal counter is reset to 0.
   (b) If the internal counter of a node representing a stem cell is equal to *CyclingPhaseAg* but not all the node's neighbors are stem cells then do nothing; leave the internal counter unchanged.
3. Empty nodes
   (a) If the internal counter at an empty node has reached *NeighborEmptyAg* and there is a stem cell neighbor then introduce, i.e., give birth to, a stem cell in that location. The internal counter of the node is reset to 0.
   (b) If the counter at an empty grid has not reached *NeighborEmptyAg* and there is exists a stem cell neighbor then increment the counter by 1.

(c) If there are no stem cell neighbors at all then reset the internal counter to 0.

A simulation of this model, together with its source code, can be found at:

http://doc.gold.ac.uk/~mas02md/cell/simulations/agur/index.html

13.6.2 Discussion about the Cellular Automata Approach

The specification of the cellular automata model reveals the following issues:

1. Empty niche spaces must do computational work, i.e., to maintain counters
2. Stem cell division is not explicitly represented, stem cells are brought into being by empty space
3. Individual stem cells are not explicitly tracked before they give rise to new stem cells
4. The state of the stem cell after division is not defined

The specification clearly reveals that niche spaces, i.e., empty nodes, must have counters for this model to work. In a sense, empty space is having to do some computational work. Clearly this lacks biological feasibility and is against what the authors state about modeling cells, rather than empty locations, having counters.

Stem cell division is not explicitly represented, instead stem cells are brought into being by empty space. More subtly, these stem cells appear when empty nodes have been surrounded by at least one stem cell for a period of time. The location of the neighboring stem cell, however, can vary at each step.

The model details the fact that if a stem cell is next to an empty space long enough then it will divide so that it's descendent occupies this space, however, the rule does not state that the neighboring stem cell must be the same stem cell for every tick of the counter. It states something much weaker; that there must be a neighboring cell, possibly different each time, for each tick of the counter, from 1 to *NeighborEmptyAg*. Biologically, it would seem more intuitive that the same stem cell should be next to an empty niche space for this length of time in order for "division" to occur into the space but the model lacks a "directional component".

The state of a stem cell after division is not defined. Let us for a moment assume that the neighboring stem cell (S) is fixed for all counts from 1 to *NeighborEmptyAg* from some specific location (N). Nothing is said about what happens to S after a new stem cell appears in N. For example, should the counter of S be reset after division? Neither does it give any preconditions on S. For example, does S's local counter need to have reached an appropriate point in its cycling phase for this to happen?

So the basic problem is that this model relies on allowing both unfilled niche locations as well as stem and determined cells to have counters. Moreover, it does not investigate or model the nature of a stem cell before and after division. We now attempt to re-interpret these rules using an agent-based approach that still retains the overall qualities of the model.

13.6.3 Re-formulation Using an Agent-Based Approach

One of the biggest differences between the original cellular automata model and our re-formulation is the change in the role of graph nodes. In the cellular automata model each node represents either a cell or an empty space. In our re-formulation, each node represents

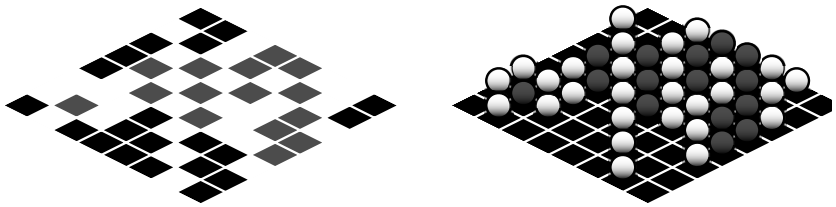

FIGURE 13.2 A comparison of the original Agur cellular automata model and our reformulation as a grid-based agent model. In the original model the nodes maintain the state of the cells, whereas in our re-formulation the nodes contain agents and it is the agents that maintain the state of the cells.

a space that may or may not contain an agent that represents a cell. This difference in the two models is illustrated in Figure 13.2.

With the agent approach we also provide each cell with a unique identifier. We model all cells as having one internal counter as before. In addition, each cell maintains a counter associated with each of its neighboring nodes. The counters associated with neighboring nodes record how long the neighboring location has been empty. Moreover, cells can sense the type of cell at each of its neighbors, although this perception ability is only used by stem cells. If an agent represents a stem cell then it can potentially divide into any location where the counter has reached *NeighborEmptyAg*.

A cell agent is specified formally as:

$$[AgentId]$$

AgentCellAg

$$\begin{aligned}
&id : AgentId \\
&type : TypeAg \\
&counter : \mathbb{N} \\
&nscounter : Node \pfun \mathbb{N} \\
&nstype : Node \pfun TypeAg \\
\hline
&type = StemAg \lor type = DeterminedAg \\
&\mathrm{dom}\, nscounter = \mathrm{dom}\, nstype \\
&\forall n : Node \mid nstype\ n \neq EmptyAg \bullet nscounter\ n = 0
\end{aligned}$$

A stem cell agent is defined as follows. Other types are defined similarly.

AgentStemCellAg

$$\begin{aligned}
&AgentCellAg \\
\hline
&type = StemAg
\end{aligned}$$

The system state consists of the niche where some nodes are filled with cells. The first predicate simply states that the empty nodes are those nodes which do not contain a cell. The second predicate states that the neighbors are defined by the graph to which the cells are attached.

AgentSystemStateAg

$$
\begin{array}{l}
cells : Node \rightarrow AgentCellAg \\
emptynodes : \mathbb{P}\, Node \\
\hline
emptynodes = Node \setminus (\mathrm{dom}\, cells) \\
\forall\, n : Node;\ c : AgentCellAg \mid (n, c) \in cells \land c.type = StemAg\bullet \\
\qquad \mathrm{dom}\, c.nscounter = \mathrm{ran}(\{n\} \lhd graph)
\end{array}
$$

13.6.4 Operation

Space does not permit us giving a full treatment, but we outline the basic operations here.

1. Cells set/update counters.
2. Mature stem cells that are surrounded by empty neighbors and have neighbor counters that have reached *NeighborEmptyAg* will make a request to the environment to divide into two daughter stem cells.
3. The environment resolves any conflicts where several cells wish to divide into the same node (region) and informs those mature stem cells that can divide and those that are not able to.
4. Mature stem cells that are able to divide do so. Mature stem cells that are surrounded by stem cells become new determined cells. Mature determined cells which are ready to leave the niche do so.

Our agent-based approach to modeling forces us to consider what happens when two stem cells attempt to divide into the same location. In our model, we specify that when the internal counter reaches *CyclingPhaseAg*, it signals to the environment the niche spaces that it is prepared to divide into.

Notice, that this approach is also agent-based in nature. Namely, the agent attempts to do something but the environment is a dynamic and uncertain one. From the perspective of a single cell with its limited sensory abilities the world is no longer deterministic like it was in the cellular automata model, and not all attempts at action will be successful.

The agent-based model not only considers the nature of acting in a dynamic environment but also addresses issues such as the basic physical limitations of the stem cell niche in general. Once again, it's difficult to see how such issues can be considered, at least explicitly, with the cellular automata approach.

A stem cell agent that is ready to divide, signals to the environment those neighbors that have been empty for long enough, and so are able to receive the new cell. Of course the output may be empty.

The environment receives requests from cells to divide, and non-deterministically assigns those cells that can divide and those that have insufficient space around them. There are several safety properties that we can specify here:

1. all agents get a reply (first predicate)
2. no agent can be told to divide and not divide (second predicate)
3. no node ever has more than one agent dividing into it (third predicate)
4. cells only get to divide into a node they have requested (fourth predicate)
5. there is no remaining empty node that has been requested by any of the agents not-granted division (fifth predicate).

Cells that divide get told where they should divide into. We have two alternatives with the assignment of identifiers to the daughter cells: we can either give both daughters new identifiers or allow one of the daughter cells to keep the Id of its parent. For the purposes of tracking the inheritance between parent and daughter cells, we prefer the first of these alternatives and give each daughter cell a unique Id.

We have run many simulations of both the original CA model and of our agent recapitulation to check that the behaviors of our agent model has the same properties of the CA model. As we explained above, the agent model has allowed us to address the issues of biological implausibility.

It is interesting to note that allowing cells to split into all available spaces, i.e., up to four daughters, gives us the closest possible agent-based simulation match to the original CA models, however, any biologically plausibility we may have introduced would be negated by this. Going back to the original CA model we see that this four-way division is supported, further reducing the plausibility of the model. By limiting cell division to result in a maximum of at most two daughter cells we still maintain the integrity of the original cellular automata version.

All these simulations are available on the web site.

13.6.5 Roeder-Loeffler Model of Self-Organization

A recent example of an approach that uses a more sophisticated model that more successfully addresses issues of biological implausibility, is that of Markus Loeffer and Ingo Roeder at the University of Leipzig, who model hematopoietic stem cells using various, but limited, parameters including representing both the growth environment within the marrow, one particular stem cell niche, and the cycling status of the cell [Roeder and Loeffler, 2002; Roeder et al., 2005; Roeder and Loeffler, 2004; Roeder et al., 2006]. In this model, the ability of cells to both escape and re-enter the niche and to move between high and low niche affinities, referred to as within-tissue plasticity, is stochastically determined.

The validity of this model is demonstrated by the fact that it produces results in global behavior of the system that match experimental laboratory observations. The point is that the larger patterns of system organization emerge from these few simple rules governing variations in niche-affinity and coordinated changes in cell cycle.

In recent work it has been shown to model cases of chronic myloid leukemia [Roeder et al., 2006]. Although the Roeder-Loeffler model is is rather sophisticated (it is formal, there is a simulation, it addresses key issues of self-organization and much of the modeling has an agent-like quality to it), there are a number of issues regarding this model that we have addressed by extending it using our agent framework. Most significantly, the use of a global probability function to control the movement of cells between environments, and in the agent-view this is problematic; this probability is calculated from global information relating to the numbers of various cells in the system. Although it useful to assume access to this global information when developing the model of stem cell behavior, no mechanism is known for how stem cells could have access to this information in real biological systems.

We will say more about this model when we discuss the implementation, but to summarize we have extended the Roeder-Loeffler model to produce an agent-based model that increases the biological intuition and plausibility of the model, and allows us to investigate emergence due to the subtle changes in micro-environmental effects for each cell. Modeling cells as agents responding autonomously to their local environment is much more fine grained than the previous model using equations to model cell transitions and allows for a much greater degree of sophistication in the possibilities of understanding how self-organization actually takes place in the adult human body. The main point is that an agent does not rely on

getting information about the system state, in keeping with the reactive multi-agent systems approach, and we believe that this gives a more biologically plausible handle on how things might be working at the micro-environmental level.

In general the agent models we produce suffer from none of the drawbacks of other formal systems approaches. However, models of agents by themselves have no real practical value as we do not have any method of determining how the individual behavior of agents affects the overall system behavior. Formality is useful for building consistent clearly defined models but they give us very little clue as to how the overall system of interacting agents will actually evolve. Therefore we need to move from modeling to simulation, and this is considered next.

13.7 From Agent Model to Simulation

To investigate the behavior of our models we have developed computational models to simulation populations of stem cells as a Multi-Agent Based Simulation (MABS). MABS is an approach to simulation that builds on Agent-Based modeling (ABM) and Multi-Agent Systems (MAS) to develop simulations of real-world phenomena as systems of multiple interacting agents. Implementing MAS and MABS models is challenging* and several simulation frameworks have been developed to facilitate this process. In this section we will describe our implementation of the agent-based model as a MABS using MASON [Luke et al., 2004], a MAS/MABS development environment.

Examples of agent modeling tools that span different approaches to modeling multi-agent systems include:

FIPA-compliant tools: The Foundation for Intelligent Physical Agents (FIPA) is the standards organization for agents and multi-agent systems. It has established a standard for inter-agent communication that is the core of the FIPA agent system model. The FIPA approach to modeling agents has been widely adopted by multi-agent system tool developers. FIPA-compliant tools include JADE [Bellifemine et al., 1999], Aglets [Lange and Mitsuru, 1998].

StarLogo: StarLogo was developed as an educational tool by Mitchell Resnick to introduce children to issues of emergence through computational modeling of *turtles* and *patches*. Where patches model the environment and turtles represent agents that inhabit patches. Since its introduction, StarLogo has inspired other environments for multi-agent simulations, most notably NetLogo [Wilensky, 1999].

Sugarscape: In 1996, Epstein and Axtell introduced Sugarscape, an initial attempt to develop a "bottom up" social science [Epstein and Axtell, 1996]. Sugarscape simulates the behavior of artificial people (agents) located on a landscape of a generalized resource (sugar). Fundamental collective behaviors including group formation, cultural transmission, combat, and trade are seen to "emerge" from the interaction of individual agents following simple local rules.

Swarm: In the Swarm multi-agent simulation platform the basic unit of simulation is the *swarm*, a collection of agents executing a schedule of actions. Swarm supports

*In the early stages of this project we considered using Erlang [Armstrong et al., 1996], a language based on Haskell, developed by Ericsson, that is well suited to modeling our agent framework but lacks a MABS environment. Sadly, developing such a framework was outside the scope of this project, although we are still very interested in using this powerful language in the future.

hierarchical modeling, such that agents can be composed of swarms of other agents. As a computational modeling approach, Swarm has inspired several other popular simulation platforms, including Repast and MASON.

Several existing multi-agent simulation tools were considered for this part of the project, including Swarm [Minar et al., 1996], NetLogo [Wilensky, 1999], Ascape [Inchiosa and Parker, 2002] and Repast [Tatara et al., 2006]. We chose a relative newcomer to the simulation frameworks, MASON, as we found that its carefully thought out architecture provided the best framework for our research.

13.7.1 MASON

MASON (Multi-Agent Simulator Of Neighborhoods/Networks) is a multi-agent simulation development environment and simulation tool [Luke et al., 2004]. MASON has been developed around a discrete-event multi-agent simulation library written in Java. This flexible foundation allows for a range of simulation approaches, including discrete step and discrete event simulations; so far, we have used it to develop discrete step simulations.

MASON was designed to be the foundation for large custom-purpose Java simulations. MASON contains both the simulation library and an optional suite of visualization tools in 2D and 3D. Features of MASON that were particularly desirable for our development, that affected our decision to use it, include:

Portability: The developers of MASON have taken great care to ensure that cross-platform portability, going as far as implementing their own version of the Merseinne-Twister algorithm for generating random numbers so that stochastic elements of simulations will run identically across machines and platforms.

Speed and size: The MASON library has been developed to be fast and relatively small, allowing many cells to be simulated at once using standard desktop and notebook computers. This has allowed us to simulate relatively large stem cells systems without the need for complex distributed computing solutions.

Separation of model and visualization: Models are completely independent from visualization, which can be added, removed, or changed at any time. This has allowed us to easily develop simulations that can be run either with a visualization for demonstration purposes, or without a visualization for experimentation.

Checkpointing: Models may be checkpointed and recovered, or dynamically migrated across platforms. This allows stem cell models to be tested by first running them on a development machine, and then migrating them to a dedicated server for long-term processing.

Embedding: Models are self-contained and can run inside other Java frameworks and applications. This allows us to embed simulations in other Java applications and applets. In addition, the MASON library sets no limits on the use of other libraries and frameworks, e.g., allowing the graphing libraries to be used to provide numerical feedback as a visualization option.

Visualization: The MASON library includes support for several standard types of 2D and 3D visualizations, using the core Java libraries for 2D and 3D graphics. These have provided a foundation for developing specific visualizations of stem cells and their environments in our models.

Export of rich media: The MASON development environment and the MASON library support the export of images (PNG) and movies (Quicktime) for docu-

menting particular simulation runs. In addition, MASON provides support for the generation of other output data streams, e.g., XML.

Advantages of Using MASON

Combining the powerful features offered by MASON has provided us with an excellent environment for development, presentation and experimentation. During development the MASON development environment allows for rapid testing and evaluation of the simulations. During presentations, the speed and size of the MASON library allow users to demonstrate "live simulations" on standard notebook hardware. During experimentation we run large-scale experiments using simulations without visualizations to automate "batch processing" of stem cell models across clusters of machines.

13.7.2 Implementation of CELL in MASON

The MASON library provides a comprehensive foundation for developing our agent foundation, the basis of which is a thin layer around the `Steppable` interface class. This is described by the Unified Modeling Language (UML) [Rumbaugh et al., 1999] *class diagram* in Figure 13.3.

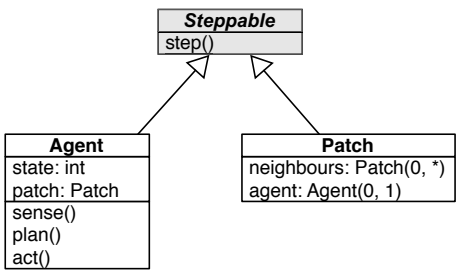

FIGURE 13.3 Our framework is rooted with implementations of `Agent` and `Patch` that implement the MASON interface `Steppable`.

The `Agent` class implements the `Steppable` interface to implement the base class for all cell classes in models. The `Agent` class decomposes the `step()` method into a sequence of three methods, `sense()`, `plan()` and `act()`, controlled by the `state` of the agent. Every instance of `Agent` can be assigned to a single instance of `Patch`. The `Patch` class implements the `Steppable` interface to provide the base class for decomposing space in simulations for the simulation of processes over the space, e.g., chemical diffusion. Each instance of `Patch` contains a set of `neighbors` that define the topology of the space.

The `Patch` class corresponds to a *region* in our framework, although the exact details of a region for a model are left to the individual implementations. In accordance with our formal framework, `Patch` is used as a base for building the environment, but apart from easing certain implementation issues its inheritance from `Steppable`, e.g., allowing the use of existing portrayals for visualization purposes, it acts as a passive element in the implemented systems, simply recording the topology of the environment, as well as dynamic features updated by the simulation engine.

The CELL framework provides a utility class for simulating the diffusion of chemicals, the `DoubleGrid2DDiffuser` class acts on two-dimensional arrays of floating point numbers,

i.e., doubles, to diffuse the values across the array. This class implements the Steppable interface as illustrated in Figure 13.4 so that it can be inserted into schedules, just like agents. In the simulations developed so far, the DoubleGrid2DDiffuser class is used to simulate the diffusion of chemicals released by stem cells and stromal cells.

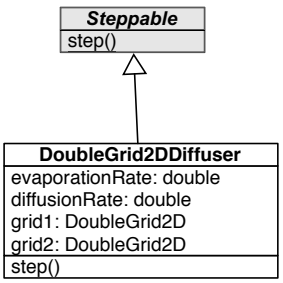

FIGURE 13.4 UML diagram of the Double2DDiffuser class.

In MASON, agents and environments cannot directly render themselves to a visualization, instead simulations are visualized using *portrayals*. The CELL framework defines two convenience portrayal classes for use by simulations, see Figure 13.5. CellPortrayal2D renders cells within a visualization as a simple oval. ChemicalGridPortrayal2D renders chemical concentrations, held in a DoubleGrid2D, using a gradient to indicate the concentration of patches. Similarly to the implementation of Agent, these classes represent thin wrappers around existing MASON classes; OvalPortrayal2D and FastObjectGridPortrayal2D.

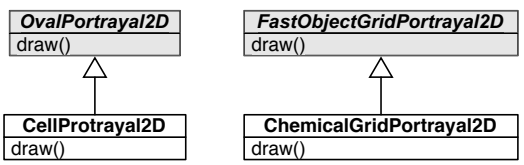

FIGURE 13.5 UML diagram of the Portrayal2D subclasses.

13.7.3 Implementation of Roeder-Loeffler Model in MASON

The small set of classes described above forms the framework, built on MASON, that we implement specific models upon. This section details the classes developed to implement the Roeder-Loeffler model of stem cell organization. This implementation adds specific details of the model of cells and their environment as described.

Figure 13.6 illustrates the implementation of the basic Cell class. The Cell class is a subclass of the Agent class and overrides the sense(), plan(), and act() methods to implement the standard agent cycle within the specific environment described for the Roeder-Loeffler model. Each Cell class maintains an internal state within an instance of the CellStateInternal class.

Cells communicate with the environment, simulation engine, visualization, etc. through the use of CellEvents that CellListeners can subscribe to be notified about. Cells will

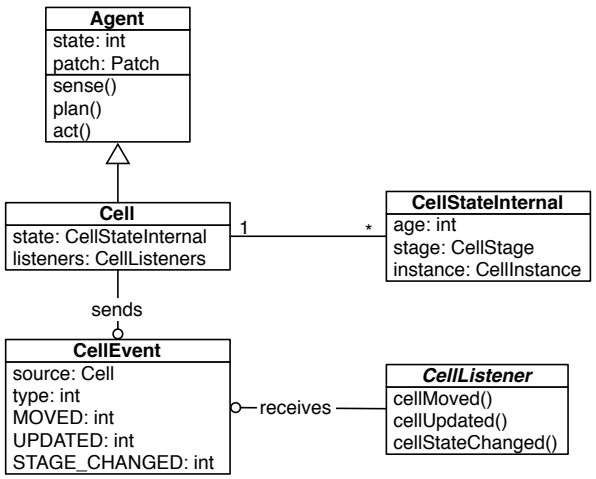

FIGURE 13.6 UML diagram of the `Cell` class.

generate events when they move (`CellEvent.MOVED`), have their internal state updated (`CellEvent.UPDATED`), or when the stage that they are in changes (`CellEvent.STAGE_CHANGED`).

Figure 13.7 illustrates the classes that implement the *cell typing* system in the Roeder-Loeffler model. The `CellType` class is the core of this system and allows different types of cells, e.g., stem cells and stromal cells, to be defined with different *cell cycles*, using the `CellCycle` class, for each *stage* in the cell's life, represented using `CellStage` objects.

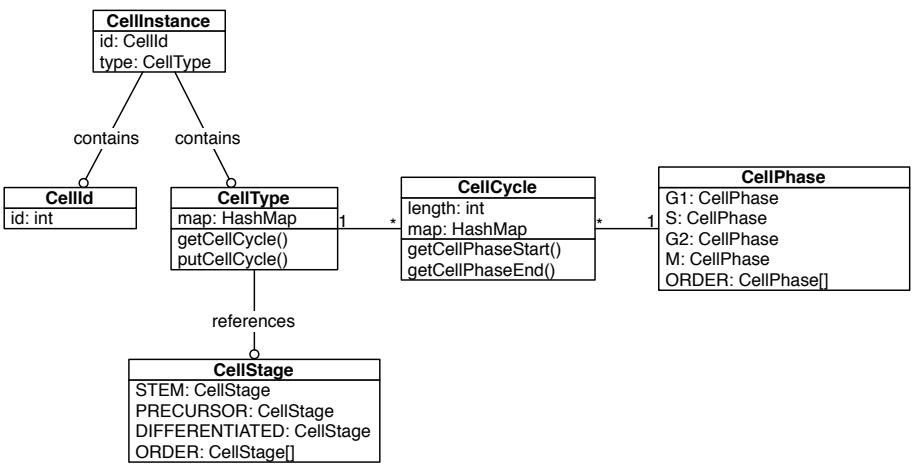

FIGURE 13.7 UML diagram of the `CellType` and related classes.

Possible `CellStage`s in this implementation Roeder-Loeffler model are `STEM`, `PRECURSOR` and `DIFFERENTIATED`. Hematopoetic cells in the simulation can potentially pass through all of these stages. Stromal cells begin in the `DIFFERENTIATED` stage and remain in it throughout simulation runs as defined in the model.

Cell cycles are defined as a sequence of *phases* that cells pass through, the different phases

are represented by the `CellPhase` class. The phases that our cells pass through are: `G1`, `S`, `G2` and `M`. These are defined in this order in the `CellPhase` class. Different types of cells are defined using the `CellType` class as spending different lengths of time in these phases.

Figure 13.8 illustrates the classes that form the top-level in the agent-based Roeder-Leoffler simulation, i.e., `WorldState` that maintains all of the information about the environment and `AgentRoeder` that implements the top-level simulation engine and stand-alone application. The `AgentRoederWithUI` class wraps the `AgentRoeder` class with a user-interface for displaying visualizations produced using the portrayal classes defined in the CELL framework. Using this pattern of developing a stand-alone application for running on the command-line without a visualization, and a wrapper to add the necessary GUI elements when desired, provides a powerful method for developing the simulations.

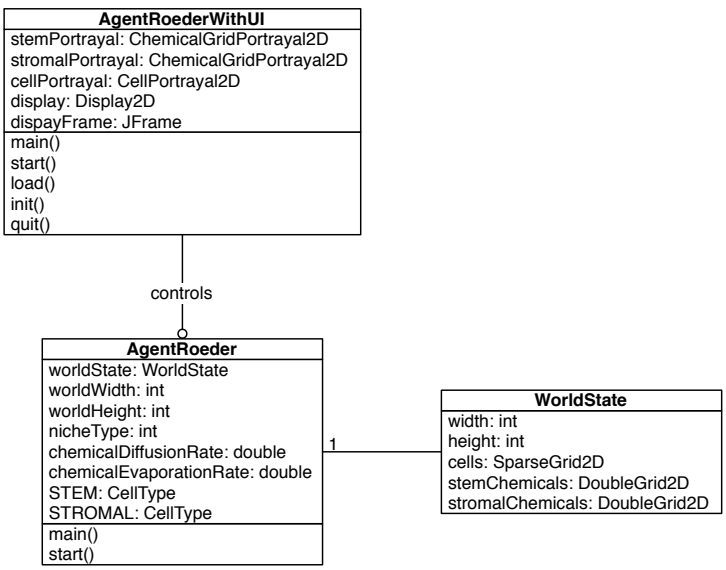

FIGURE 13.8 UML diagram of the `WorldState` container class, the `AgentRoeder` simulation class and the `AgentRoederWithUI` visualization class.

Using the sophisticated library of components provided by MASON has allowed us to build an agent framework comprised of some utility classes, and implement specific models with relatively little code.

13.8 Discussion

We have spoken much in this chapter about why agents are "good" for modeling natural systems but perhaps the best way to see the advantages is in the visualizations of the simulations themselves. Again we encourage the reader to look at the visualizations for themselves on the project web site*, but we present some of the images here.

*The CELL Project web site is located at http://doc.gold.ac.uk/~mas02md/cell/

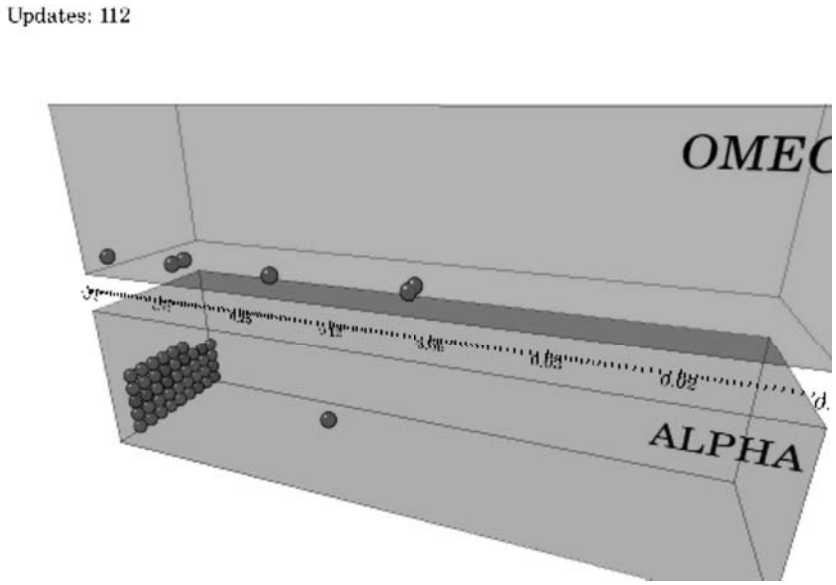

FIGURE 13.9 The visualization of the original model. We see cells switching between two compartments according to the calculation of a probability function.

We have extended the Roeder model to incorporate a model of space so that we can consider cell movement in more detail. The difference between the two approaches can best be seen by looking at the visualizations online.

In Figure 13.9 we see a visualization of the Roeder-Loeffler model, based on an existing visualization, where cells are in one of two compartments. The compartments represent broad environments inside (α) and outside (Ω) the stem cell niche. Movement within the compartments represents changes in niche affinity, as indicated by the scale, not physical movement within the environments. Consequently, the visualization should be read more like a 3D graph than a representation of 3D space. When the visualization moves in time, we see cells switching between the two compartments. The particular simulation being shown is a competition between two populations of stem cells, red and blue, competing for control of the niche. The biological intuition behind what is actually happening is very difficult to unpick.

In our agent view, shown in Figure 13.10, we get an understanding of the actual *mechanics* of what is happening to individual systems and how the system as a whole is interacting. Movements of agents in the simulated environment are represented as movements in the visualization. The secretion of chemicals from stem cells and stromal (niche) cells is what keeps this system continually generating new blood cells and maintaining the population of stem cells. Chemical concentrations are represented in the visualization by the coloring of regions according to the chemicals they contain. From this visualization, it is clear that we can experiment with properties such as the physical shape of the stem cell niche by changing the placement of stromal cells, something that isn't directly possible with the original model. The real value of visualizations like this is that biologists are attracted to them.

One of the predictions that our agent-based visualization makes is that stem cell activity pulses around the niche. Although not verified to date it is a testable prediction made about the nature of stem cell activity. We are currently seeking interested biologists to investigate whether this prediction can be verified.

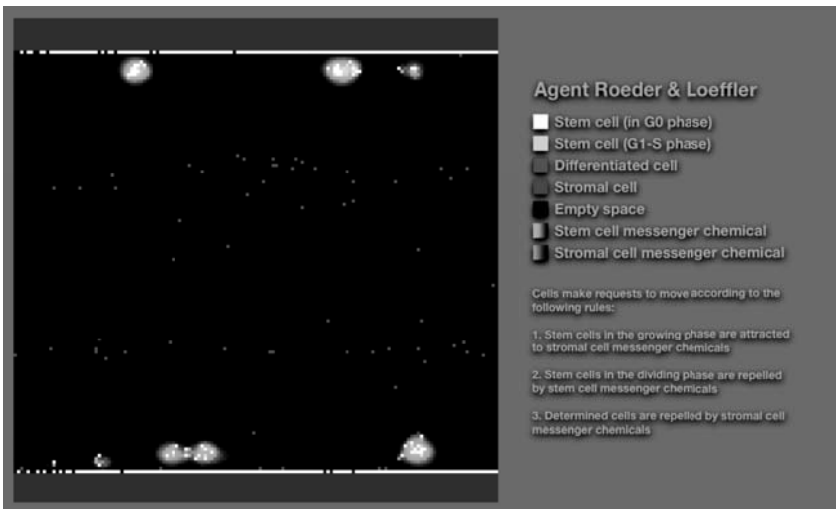

FIGURE 13.10 The visualization of our agent model. In our view a much more biologically intuitive visualization.

Our next goal is to build an agent-based model of chronic myeloid leukemia which will aim at providing an opportunity to evaluate the *mechanisms* which are taking place when this cancer takes hold in the development of blood cells.

13.8.1 Concluding Remarks

Recent medical evidence suggests that the way to understand how stem cells organize themselves in the body is as a self-organizing system, whose global behavior is an emergent quality of the massive number of interactions of cells with each other and of the environment of which they are a part. We claim, therefore, that the multi-agent system approach to modeling is the most suitable one for exploring means to simulate the behavior of stem cells and from resulting simulations [Theise and d'Inverno, 2004; d'Inverno et al., 2006], suggest how tiny changes in individual stem cell behavior might lead to disease at the global, and hence observable from an experimental perspective, system level. We have outlined the benefits of this approach by comparing it to a cellular automata approach in detail.

Modeling and simulating stem cells promises much. Our intention is to build a *common conceptual framework* from which integrated research can ensue. We believe that the *agent* approach to modeling, coupled with the natural feel of the visualizations makes it a valuable currency in this effort.

Indeed, the fact that the work came from an interdisciplinary project (rather than, say, a team of mathematicians working in isolation) means that this work is ideally placed to be the foundation from which we can produce this common conceptual framework so important to harnessing the energies of research from different fields.

One last comment. Perhaps the greatest evidence for the impact of our work has been in the language used by one of our collaborators on a recent paper in *Nature* [Theise, 2005].

> Cells fulfill all the criteria necessary to be considered agents within a complex system: they exist in great numbers; their interactions involve homeostatic, negative feedback loops; and they respond to local environmental cues with limited stochasticity ('quenched disorder'). Like any group of interacting individuals fulfilling these criteria, they self-organize without external planning. What emerges

is the structure and function of our tissues, organs and bodies.

Until working on this project the stem cell researcher Theise new nothing of agents. Now they have become the currency in which he conceptualizes them.

Acknowledgments

Andrea Omicini, Neil Theise, Jane Prophet, Peter Ride, EPSRC, AHRC and the Wellcome Trust.

References

Z. Agur, Y. Daniel, and Y. Ginosar. The universal properties of stem cells as pinpointed by a simple discrete model. *Journal of Mathematical Biology*, 44(1):79–86, 2002.

J. Armstrong, R. Virding, C. Wikström, and M. Williams. *Concurrent Programming in ERLANG*. Prentice Hall, New Jersey, 2nd edition, 1996.

E. Attar and D. Scadden. Regulation of hematopoietic stem cell growth. *Leukemia*, 18 (11):1760–1768, November 2004. URL http://dx.doi.org/10.1038/sj.leu.2403515.

F. Bellifemine, A. Poggi, and G. Rimassa. JADE - a FIPA-compliant agent framework. In *Proceedings of PAAM'99*, pages 97–108, April 1999.

H. M. Blau and B. T. Blakely. Plasticity of cell fate: insights from heterokaryons. *Semin. Cell Dev. Biol.*, 10(3):267–272, 1999.

R. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1-3):139–159, 1991.

E. Conrad and J. Tyson. Modeling Molecular Interaction Networks with Nonlinear Ordinary Differential Equations. In Z. Szallasi, J. Stelling, and V. Periwal, editors, *System Modeling in Cell Biology - From Concepts to Nuts and Bolts*, chapter 6, pages 97–123. MIT Press, Cambridge, MA, 2006.

H. De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

L. Dematte, C. Priami, and A. Romanel. The blenxlanguage: A tutorial. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 313–365. Springer, 2008. ISBN 978-3-540-68892-1. URL http://dblp.uni-trier.de/db/conf/sfm/sfm2008.html#DemattePR08.

M. d'Inverno and M. Luck. *Understanding Agent Systems (Second Edition)*. Springer; Berlin, 2004.

M. d'Inverno and R. Saunders. Agent-based modelling of stem cell organisation in a niche. In S. A. Brueckner, G. D. Marzo, A. Karageorgos, and R. Nagpal, editors, *Engineering Self-Organising Systems : Methodologies and Applications*. Lecture Notes in Artificial Inteligence, Springer, Berlin, 2006.

M. d'Inverno, M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge. The dmars architechure: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems*, pages 5–53, 2004.

M. d'Inverno, N. Theise, and J. Prophet. Mathematical modelling of stem cells: a complexity primer for the stem cell biologist. In C. Potten, J. Watson, R. Clarke, and A. Renehan, editors, *Tissue Stem Cells: Biology and Applications*. Marcel Dekker, New York, 2006.

J. Epstein and R. Axtell. *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Institute Press, Washington, DC, 1996.

D. T. Gillespie. Simulation methods in systems biology. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 125–167. Springer, 2008. ISBN 978-3-540-68892-1. URL http://dblp.uni-trier.de/db/conf/sfm/sfm2008.html#Gillespie08.

M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer, Berlin, 2008. ISBN 978-3-540-68892-1. URL http://dblp.uni-trier.de/db/conf/sfm/sfm2008.html#HeinerGD08.

M. Inchiosa and M. Parker. Overcoming design and development challenges in agent-based modeling using ASCAPE. *Proceedings of the National Academy of Sciences*, 99(90003):7304–7308, 2002. doi: 10.1073/pnas.082081199. URL http://www.pnas.org/cgi/content/abstract/99/suppl_3/7304.

M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 28:27–30, 2000.

S. Kauffman. *The origins of order*. Oxford University Press, New York, 1993.

D. Lange and O. Mitsuru. *Programming and Deploying Java Mobile Agents Aglets*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998. ISBN 0201325829.

M. Loeffler and I. Roeder. Tissue stem cells: Definition, plasticity, heterogeneity, self-organization and models - a conceptual approach. *Cells Tissues Organs*, 171(1): 8–26, 2002.

M. Loeffler and I. Roeder. Conceptual models to understand tissue stem cell organization. *Curr. Opin. Hematol.*, 11(2):81–87, 2004.

S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan. MASON: A new multi-agent simulation toolkit. In *Proceedings of the 2004 SwarmFest Workshop*, 2004.

E. Merelli, G. Armano, N. Cannata, F. Corradini, M. d'Inverno, A. Doms, P. Lord, A. Martin, L. Milanesi, S. Möller, M. Schroeder, and M. Luck. Agents in bioinformatics, computational and systems biology. *Briefings in Bioinformatics*, 8(1): 45–59, 2007. URL http://dx.doi.org/10.1093/bib/bbl014.

N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The swarm simulation system: a toolkit for building multi-agent simulations. Working Paper 96-06-042, Santa Fe Institute, Santa Fe, 1996.

S. Montagna, A. Omicini, A. Ricci, and M. d'Inverno. Modelling hematopoietic stem cell behaviour: An approach based on multi-agent systems. In F. Allgöwer and M. Reuss, editors, *2nd Conference "Foundations of Systems Biology in Engineering" (FOSBE 2007)*, pages 243–248, Stuttgart, Germany, 9–12 Sept. 2007. Fraunhofer IBR Verlag. ISBN 978-3-8167-7436-5.

S. Montagna, A. Ricci, and A. Omicini. A&A for modelling and engineering simulations in Systems Biology. *International Journal of Agent-Oriented Software Engineering*, 2(2):222–245, 2008. ISSN 1746-1375. doi: 10.1504/IJAOSE.2008.017316. URL http://www.inderscience.com/search/index.php?action=record&rec_id=17316. Special Issue on Multi-Agent Systems and Simulation.

N. Nicola and G. Johnson. The production of committed hemopoietic colony-forming cells from multipotential precursor cells in vitro. *Blood*, 60(4):1019–1029, October 1982.

J. Novak and C. Stewart. Stochastic versus deterministic in haemopoiesis: what is what?

British Journal of Haematology, 78(2):149–154, 1991.

M. Ogawa. Stochastic model revisited. *International Journal Hematology*, 69(1):2–5, 1999.

C. S. Potten and M. Loeffler. Stem cells: attributes, cycles, spirals, pitfalls and uncertainties. lessons for and from the crypt. *Development*, 110(4):1001–1020, 1990.

P. Quesenberry, H. Habibian, M. Dooner, C. McAuliffe, J. F. Lambert, G. Colvin, C. Miller, A. Frimberger, and P. Becker. Physical and physiological plasticity of hematopoietic stem cells. *Blood Cell. Mol. Dis.*, 27(5):934–937, 2001.

A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Pacific Symposium on Biocomputing*, pages 459–470, 2001.

I. Roeder. Quantitative stem cell biology: computational studies in the hematopoietic system. *Curr. Opin. Hematol.*, 13(4):222–228, 2006.

I. Roeder and M. Loeffler. A novel dynamic model of hematopoietic stem cell organization based on the concept of within-tissue plasticity. *Exp. Hematol.*, 30(8): 853–861, 2002.

I. Roeder and M. Loeffler. Simulation of hematopoietic stem cell organisation using a single cell based model approach. In A. Deutsch, J. Howeard, M. Falke, and W. Zimmermann, editors, *Mathematics and Biosciences in Interaction*, pages 287–293. Birkhaeuser Verlag, Berlin, 2004.

I. Roeder, L. M. Kamminga, K. Braesel, B. Dontje, G. d. Haan, and M. Loeffler. Competitive clonal hematopoiesis in mouse chimeras explained by a stochastic model of stem cell organization. *Blood*, 105(2):609–616, 2005.

I. Roeder, M. Horn, I. Glauche, A. Hochhaus, M. C. Mueller, and M. Loeffler. Dynamic modeling of imatinib-treated chronic myeloid leukemia: functional insights and clinical implications. *Nat. Med.*, 12(10):1181–4, 2006.

A. W. Roscoe. *The Theory and Practice of Concurrency*. Prentice Hall International, New Jersey, 1997. ISBN 0136744095.

J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley Professional, 2nd edition, 1999. ISBN 0201309980.

R. Schofield. The relationship between the spleen colony-forming cell and the haemopoietic stem cell. *Blood Cells*, 4(1-2):7–25, 1978.

C. Smith. Hematopoietic stem cells and hematopoiesis. *Cancer Control*, 10(1):9–16, Jenuary/February 2003.

V. Sowmya and P. W. Zandstra. Toward predictive models of stem cell fate. *Cyotechnology Review*, 41(2):75–92, 2003.

J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall International Series in Computer Science, 2nd edition, 1992. ISBN 0139785299.

C. L. Talcott. Pathway logic. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 21–53. Springer, Berlin, 2008. ISBN 978-3-540-68892-1. URL http://dblp.uni-trier.de/db/conf/sfm/sfm2008.html#Talcott08.

E. Tatara, M. North, T. Howe, N. Collier, and J. Vos. An introduction to repast modeling by using a simple predator-prey example. In *Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects*. Argonne National Laboratory, Argonne, IL USA, 2006.

N. Theise. Now you see it, now you don't. *Nature*, 7046(435):1165, June 2005.

N. D. Theise and M. d'Inverno. Understanding cell lineages as complex adaptive systems. *Blood, Cells, Molecules and Diseases*, 32:17–20, 2004.

G. Theodoropoulos, R. Minson, M. Lees, and R. Ewald. Simulation Engines for Multi-Agent Systems. In A. Uhrmacher and D. Weyns, editors, *Agents, Simulation and Applications*. Taylor and Francis, Boca Raton, FL, 2008.

I. Thornley, R. Sutherland, R. Wynn, R. Nayar, L. Sung, G. Corpus, T. Kiss, J. Lipton, J. Doyle, F. Saunders, S. Kamel-Reid, M. Freedman, and H. Messner. Early hematopoietic reconstitution after clinical stem cell transplantation: evidence for stochastic stem cell behavior and limited acceleration in telomere loss. *Blood*, 99 (7):2387–2396, 2003.

J. Till, E. Mcculloch, and L. Siminovitch. A stochastic model of stem cell proliferation, based on the growth of spleen colony-forming cells. *Proc Natl Acad Sci USA*, 51 (1):29–36, 1964.

J. Trentin. Influence of hematopoietic organ stroma (hematopoietic inductive microenvironment) on stem cell differentiation. In A. Gordon, editor, *Regulation of Hematopoiesis*, volume 1, pages 161–168. Appleton-Century-Crofts, New York, 1970.

A. M. Uhrmacher, D. Degenring, and B. Zeigler. Discrete event multi-level models for systems biology. In C. Priami, editor, *Transactions on Computational Systems Biology I*, volume 3380 of *Lecture Notes in Computer Science*, pages 66–89. Springer, Berlin, 2005. ISBN 3-540-25422-6.

H. E. Wichmann and M. Loeffler. *Mathematical modeling of cell proliferation: Stem cell regulation in hemopoiesis.*, volume 1,2. CRC Press, Boca Raton, 1985.

U. Wilensky. *NetLogo*. http://ccl.northwestern.edu/netlogo/, 1999.

A. Wilson and A. Trumpp. Bone-marrow haematopoietic stem-cell niches. *Nature Reviews Immunology*, 6(2):93–106, February 2006.