Exploratory Modeling of Complex Information Processing Systems

Jochen Kerdels and Gabriele Peters
Chair of Human-Computer Interaction, University of Hagen, Universitätsstrasse 1, D-58097 Hagen, Germany

Keywords: Exploratory Modeling, Modeling of Complex Systems, Emergence, Self Organization.

Abstract: A widely adopted approach to study and understand complex systems such as ant colonies or economic systems consists in their modeling and simulation. In contrast to the predominat use of models and simulation in science as a substitute for the real system in order to predict the system's beaviour, the methodology of *exploratory modeling* uses modeling and simulation as a tool to increase knowledge and understanding of the systems themselves, for example to better understand the dynamic properties of a system. In this paper we propose a model which was specifically devised to support this process of exploratory modeling. The model defines four lightweight building blocks such as information processing entities that can be freely combined to model a particular complex system. Furthermore, the model provides an explicit state representation that comprises the entire model *including* an explicit representation of the information that is individually available to every information processing entity of the model. We illustrate our introduction of the proposed model by means of short examples of a concrete system for the simulation of motions in a flock of birds.

1 INTRODUCTION

A widely adopted approach to study and understand complex systems consists in their modeling and simulation. Typical examples of systems that are being investigated in that way include ant colonies, economic systems, nervous systems or biological evolution (Newman, 2011). Yet, much simpler systems based on only a small set of fixed rules like board games or cellular automata are also analyzed in this context (Evans, 2001; Holland, 1998; Gardner, 1970).

In contrast to the predominat use of models and simulation as a means to make valid predictions about the system that is being modeled, the analysis of the systems themselves typically uses the methodology of *exploratory modeling* (Bankes, 1993). Exploratory modeling is an iterative process which uses modeling and simulation as a means to test assumptions, to uncover unexpected implications of existing knowledge and to act as a form of *intuition pump* (Dennett, 1998).

At the center of this approach lies the concept, that a system is not represented by a single specific model but rather by a *model space* which is constrained by what is known about the system and which is expanded by what is uncertain. This model space is then explored by conducting series of simulations with varying model instances drawn from the space of possible models. These *computational experiments* can support the analysis of complex systems in different ways. For example:

- *Exploration of Parameters* The influence of model parameters on the dynamic behaviour of a complex system can be estimated by identifying the parameter ranges that are essential for a stable operation of the system or, conversely, by identifying those parameter ranges under which the system breaks down.
- *Identification of common Patterns*. Simulation of a wide range of complex systems facilitates the identification of common patterns that underlie phenomena of self-organization and emergence. These common patterns are the prerequisite for a more general understanding of the dynamic behaviour of complex systems (Holland, 1998).
- *Existence Proofs*. Modeling of hypothetical, high level mechanisms or structures that are based on known basic mechanisms of a system can provide proof of concept explanations for observed phenomena that are associated with that system.

The methodology of exploratory modeling is not bound to a specific form of modeling or simulation. The types of models used range from simple sets of mathematical equations (Lempert et al., 1996) to cellular automata (Langton, 1986) or agent-based mod-

Kerdels J. and Peters G..
Exploratory Modeling of Complex Information Processing Systems.
DOI: 10.5220/0004598205140521
In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2013), pages 514-521
ISBN: 978-989-8565-70-9

els[1] (DeAngelis and Mooij, 2005). The choice of a particular model type is commonly guided by the similarity between characteristics of the system and the model. For example, agent-based models are widely used in social sciences (Salamon, 2011) and economics (Bonabeau, 2002) as the concept of an *agent* readily translates into entities of the respective systems like *market participants*, *employers* or *friends*.

This direct mapping of system entities onto model entities facilitates an intuitive way of describing a system. However, this intuitive approach also emphasizes two detrimental tendencies in the resulting models:

- There is a tendency to focus on modeling the entities of a system rather than a focus on modeling the interactions occuring in that system.
- There is a tendency to incorporate only a small number of temporal and spatial scales which leads to either oversimplified entities or to entities which are complex systems by themselves. Some model types, for example the *swarm* model (Minar et al., 1996), mitigate this tendency by allowing nested structures.

Especially in the case of complex systems that employ a substantial variety of signaling mechanisms across several spatial and temporal scales, e.g. neuronal processes, these two tendencies can lead to a model that does not represent the particular system well and thus impedes the analysis of the system. In the following sections we propose a general model for the description of complex information processing systems that addresses theses issues by emphasizing the modeling of interactions and employing a description of the system that is based on the composition of building blocks which represent functional characteristics rather than discrete entities.

2 OVERVIEW AND OBJECTIVES OF OUR APPROACH

In order to support the methodical approach addressed in the previous section we designed a general model to describe complex information processing systems. This approach makes the assumption, that every complex system can be represented by a corresponding complex information processing system, i.e., that the complex system can be reduced to a system that only consists of a set of entities that process and exchange information. The model we present here was designed particularly with regard to the following five objectives:

1. *Generality*. It should be possible to model a diverse set of complex systems that would otherwise be modeled by spezialized types of models like cellular automata, neuronal networks or agent-based systems. The ability to model a wide spectrum of different complex systems is a prerequisite for the *identification of common patterns* among those systems.
2. *Explicit State Representation*. The model of a particular system should have an explicit representation of it's state at any time step resulting in an explicit representation of the models behaviour as a sequence of states. Such a representation is particularly useful for comparing the behaviour of individual models drawn from the model space of a complex system.
3. *Focus on Interactions*. The model should facilitate the description of complex interactions occuring in a system including aspects of nontrivial addressing and temporal as well as spatial propagation of information. Complex interactions are reckoned as key contributing factor to phenomena of self-organization and emergence (Holland, 1998).
4. *Expressiveness*. The perceived concepts of a complex system, i.e., its different entities, their possibly dynamic relations, and further characteristics of interest, should intuitively translate into corresponding elements of the model. The structure of the resulting models should be highly modular. A modular structure supports the variation of individual model elements and thus facilitates the creation of a broader model space beyond the mere variation of numerical parameters.
5. *Minimalism*. The interference between concepts and structure inherent to the model itself and those perceived of the complex system that is modeled should be as minimal as possible, i.e., the model should mitigate the phenomenon known as "Maslow's Hammer" (Maslow and Wirth, 1966).

The subsequent description of the model is structured as follows: Section 3 introduces the four major building blocks of the model and describes how theses building blocks relate to one another and how they can be combined. Section 4 explains how time is represented in the model, it defines what comprises a state and describes how state transitions are performed. Finally, section 5 reviews the presented model with respect to the objectives stated above.

To illustrate the description of the proposed model we will roughly sketch out the modeling of a simple *boid* simulation as an example along the way. Boids (bird-like objects) were introduced by (Reynolds, 1987) as

[1] sometimes also called *individual-based* models.

a simple model of flocking behavior. In essence, the movement behavior of a boid is governed by three simple rules[2]:

1. *Separation*. Steer to avoid crowding local flockmates.
2. *Alignment*. Steer towards the average heading of local flockmates.
3. *Cohesion*. Steer towards the average position of local flockmates.

In addition to these basic rules the model can be extended with similar rules to avoid obstacles or to stay in a certain region. Despite this very simple set of rules a flock of boids can exhibit a surprising variety of motion patterns.

3 BUILDING BLOCKS OF OUR MODEL

The proposed model uses an object-oriented approach. It's main idea is to model a complex system as a combination of four basic building blocks (*proceties*[3], *messages*, *address filters*, and *procety attributes*) which will be described in detail in this section. The building blocks can be freely combined to form descriptive elements that capture specific characteristics of a complex system that do not necessarily have to coincide with individual entities of that system. In this regard our approach differs from common modeling approaches like cellular automata or agent-based models[4] which typically divide a complex system into a set of disjunct, structural entities like cells or agents and then assess and attribute their particular functions afterwards. Contrary, in our approach the functional attributes of a complex system are identified first and then combined in the form of building blocks into corresponding structural elements of the model.

The first building block of the proposed model are information processing entities, or *proceties* in short. A procety is any element of a complex system that is able to send and receive information, to create new proceties and remove existing ones. In case of the boid simulation, each boid can be interpreted as a procety: it has to receive information about the positions and headings of nearby flockmates and it has to send its own position and heading. Less obvious, possible obstacles in a boid simulation would also be modeled as proceties that actively send out information about their presence. As there is no common representation of an environment in our model, even "passive" elements have to be modeled explicitly as sources of information, i.e., proceties, if they are to be perceived by other entities of the system. At first glance this property of the model may seem to be a serious drawback, but contrariwise, this property helps to achieve the second objective stated in section 2.

The second building block are *messages*. They encapsulate the information that is exchanged between proceties. The information that is transferred by a message can be arbitrary. The recipients of a message are determined by its address. The address has the form of a set expression which allows to describe the recipients of a message on an abstract and conceptual level. It is a key component of the model in order to achieve the third objective stated in section 2.

The set expression used to specify the recipients of a message is composed of common set operators like union or intersection, and sets of proceties that are filtered by *address filters* – the third building block of our model. If an address filter is applied to a set of proceties the filter decides for every procety in the set if the procety should remain in the set or not. The information on which the address filter bases its decision is provided by a set of *procety attributes* that are exhibited by the individual proceties. Procety attributes are the last building block of the proposed model. They provide a way to describe attributes that are shared among a set of proceties. In case of the boid simulation such a shared attribute could be the position and the heading of a boid. Based on this information, a corresponding address filter could then select all boids that are within a certain distance of the boid that originated the message – effectively restricting the flow of information from one boid to its local neighborhood only. In general, shared attributes are the basic prerequisite for the definition of global relations among proceties. These global relations can then be represented as address filters and as such be used in a set expression to specifiy the recipients of a message.

In addition to this addressing scheme, messages feature another important property. They can have an arbitrary long *time to live* (TTL) once they were sent by a procety. This property is the basis for a broad variety of time-dependent interactions between proceties. As a simple example, the messages sent by "passive" proceties like the aforementioned obstacles in the boid simulation can have an unlimited TTL and thus must be sent only once. More sophisticated uses are achieved in combination with appropriate, time-

[2] See also http://www.red3d.com/cwr/boids/ for a detailed description.

[3] short form of *information processing entities*.

[4] For an overview of agent-based models see (Salamon, 2011; Allan, 2010; Railsback et al., 2006).

dependent address filters. For example, the propagation of a message over time can be described by an address filter that selects different sets of recipients depending on the age of the message, i.e., depending on how far the message has "travelled". In the boid simulation we could use this to model a message of type "boid cry" that expands radially over time from the boid that uttered the cry.

The four building blocks of the proposed model can be summarized as follows:

1. *Proceties* are "information processing entities" of a complex system. As such they process information which they send and receive in form of *messages*. Furthermore, proceties can create new proceties and delete existing ones.
2. *Messages* represent information that is transferred between proceties. The address of a message has the form of a set expression which allows to specify the recipients of a message on a high, conceptual level. Furthermore, messages have a TTL (time to live). That means, they can exist independently of any procety for an arbitrary number of time steps after they have been sent.
3. *Address filters* are used as a mechanism in the message addressing scheme of the model. They represent global relations among sets of proceties that share one or more procety attributes.
4. *Procety Attributes* represent common attributes that are shared among a set of proceties. They can be used by address filters to determine if the owner of an attribute should receive a particular message or not.

As stated at the beginning of this section, the described building blocks of our model do not necessarily have an exclusive one-to-one relationship to entities of the complex system that is being modeled. Instead, an element of a complex system can be represented by several building blocks at once. For example, the procety attribute of a procety could itself be a procety in its own right that modifies the attribute values according to the information it receives. A practical example could be the model of a biological neuron with its ion channels. In this case, the neuron as well as the ion channels could be modeled as proceties, where at the same time the ion channels would also be procety attributes of the neuron. In such a configuration, the ion channels could independently alter their behavior in reaction to ion channel specific signals, i.e., messages, that are directly processed by the ion channels themselfes and not by the neuron as a proxy.

4 TIME, STATES, AND STATE TRANSITIONS

The proposed model operates on a discrete time scale with steps $t \in \mathbb{N}$. At the beginning of each time step t the *state* of the model is given by the current states of all building blocks in the current model. This state comprises the states of all proceties, messages, address filters and procety attributes. The transition of the state at time t to the state at time $t+1$ is performed by the following three substeps:

1. Use the *procety scheduler* of the model to generate a processing schedule that governs which proceties are executed in substep 2.
2. Prompt each procety in the processing schedule to process the messages in their local message buffers.
3. Evaluate the addresses of all messages and distribute the messages according to the resulting procety sets to the local message buffers of each procety.

The use of a *procety scheduler* in the first step of the state transition allows to precisely control, how the proceties are updated during the state transition. The default procety scheduler of the model facilitates the synchronous updating of all proceties, i.e., every procety is prompted to process its messages in every time step. However, there are several models for complex systems that prefer asynchronous updating, e.g., many agent based models (Caron-Lormier et al., 2008). In these cases a custom procety scheduler can be defined to accurately emulate the updating procedure of the particular model.

The separation of the processing of messages in step 2 and the delivery of messages in step 3 effectively implements a double buffering scheme. This means that messages that were sent at time t will be processed earliest at time $t+1$. It also guarantees, that the order of the proceties inside the processing schedule has no effect on the behavior of the model.

5 REVIEW OF THE OBJECTIVES

The proposed model was specifically designed to support the process of exploratory modeling. In this regard we defined five objectives to guide the development of the model. In this section we review the model in relation to these objectives.

Generality. There are two main components of the model that provide the ability to model a wide variety of different complex system. First, the addressing scheme based on procety attributes and address filters facilitates the description of virtually any kind of static or dynamic relation between the elements of a system. Secondly, the state representation of the model provides an implicit double buffering scheme which, in combination with a customizable procety scheduler, allows to implement synchronous as well as asynchroneous updating during state transitions.

Explicit State Representation. As described in section 4 the state of a system comprises all building blocks of the model. In addition, the information that is individually available to a procety in that time step is represented by the messages inside the local message buffer of that procety. This state information is especially usefull, e.g., to extract a communication graph for every time step that represents which proceties exchange information with one another. This graph representation opens up the possibility to use common graph measures[5] to characterize the dynamic structure of the interactions occuring in the complex system. As this representation is independent of the particular system that is being modeled, it facilitates the comparison of different complex systems and thus supports the process of exploratory modelling.

Focus on Interactions. As already stated above, the addressing scheme provides an effective way to describe a wide variety of interactions between the proceties. Furthermore, the ability of messages to exists for arbitrary long periods of time and the possibility to define time-dependent address filters yields a whole new area of interactions that can be modeled, e.g., the spatial propagation of signals. The modulation of messages was not explicitly addressed. However, as the building blocks of our model can be freely combined, a message can also be a procety or an address filter for instance. The latter, for example, would allow for the manipulation of the message content while it is propagated over time.

Expressiveness. The building blocks of the model were designed in the spirit of object-orientation to encourage the encapsulation of local knowledge. For example, when proceties interact by exchanging messages they use a set expression for addressing other proceties. This set expression contains only address filters which typically represent high level concepts about some relation between the proceties. Thus, a procety does not need to know much about the other proceties they are interacting with. The same holds true for the address filter itself. The filter just relies on the information provided by specific procety attributes.

[5] An extensive review of graph measures can be found in (Newman, 2003).

Minimalism. The model uses only four building blocks which represent functional rather than structural characteristics of a complex system and which have a very precise and small "conceptual footprint". As the building blocks can be freely combined to form custom model elements there is no need to "forcefully" fit a perceived concept of a particular complex system onto a single, non-custom building block of the model.

In the context of exploratory modeling the use of building blocks leads to a highly modular structure which allows to expand the *model space* by providing alternative versions of specific model elements. For example, an address filter that determines the recipients of a message could have a probabilistic counterpart which selects the recipients of a message only with a certain probability.

6 SIMULATION EXAMPLE

We created a software library using C++ and the ROOT data analysis framework developed at CERN (Brun and Rademakers, 1996) to support the implementation of simulations that use our model. As the ROOT framework is designed to handle and analyse large amounts of data, it is well suited for the data intensive process of exploratory modeling.

As a first test case for our model we implemented a boid simulation as described in section 2. In addition to the three basic rules of a boid we added a rule that keeps the boids confined to a local area. The primary use of this model was to check the implementation of our library for errors and to gather a first set of simulation data that could be analysed within the ROOT framework.

To illustrate one application of exploratory modeling, i.e. the *exploration of parameters* as described in section 1, we investigated the influence of the parameter *viewing range* of the individual boids on the observable flocking behaviour. Therefor, we conducted a series of simulations in which we increased the viewing range stepwise from 5 units to 90 units. As a measure of "flocking" we used the average number of clusters that occured within 1000 time steps.

Is the viewing range to short, e.g. only 10 units, then no observable flocking occurs (s. Figure 1). If the viewing range is increased to some medium value, e.g. 35 units, then the emergence of several independent flocks can be witnessed (s. Figure 2). However, if the viewing range is increased further, e.g. to a value of 80 units, then the dynamic behavior breaks down and all boids accumulate in a single cluster (s. Figure 3). Figure 4 summarizes the results of these

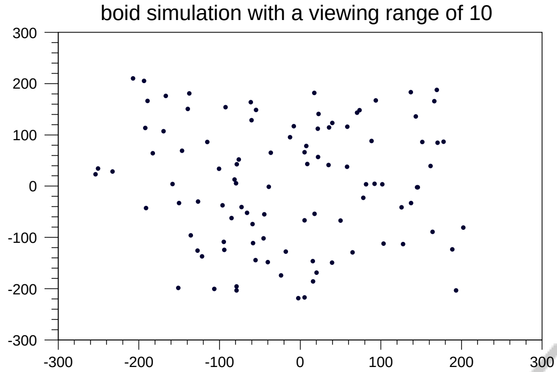

Figure 1: A representative distribution of boids with a short viewing range of 10 units.

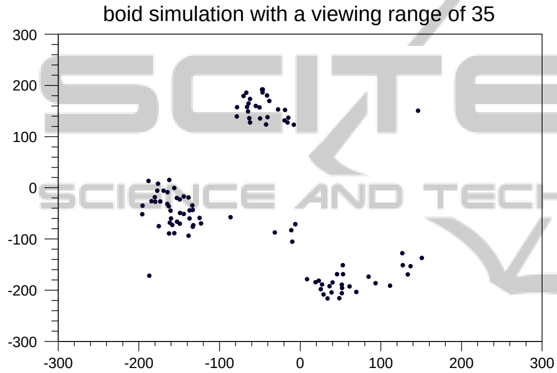

Figure 2: A representative distribution of boids with a medium viewing range of 35 units.

experiments. With increasing viewing range the number of clusters drop rapidly starting with ~ 100 clusters at a viewing range of 5 units to ~ 10 clusters at a viewing range of 35 units. Starting with a viewing range of 60 units the simulation averages on one single cluster, i.e. no flocking occurs anymore.

This simple example illustrates one application of

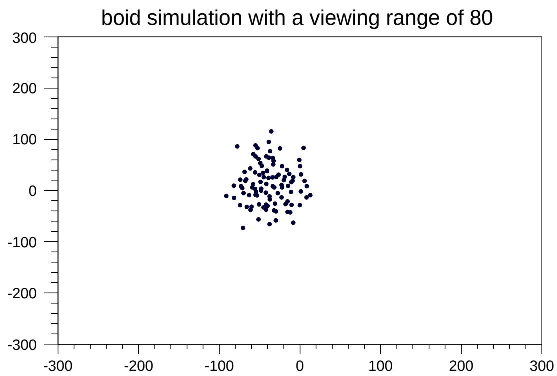

Figure 3: A representative distribution of boids with a long viewing range of 80 units.

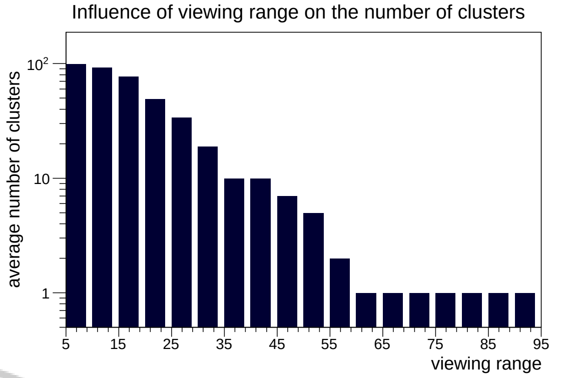

Figure 4: The viewing range of the individual boids influences the average number of clusters that emerge during a simulation run.

exploratory modeling to increase the understanding of the dynamic properties of a given complex system. It provides a first intuition about how the parameter *viewing range* influences the overall dynamic behavior of the system. Yet, a full exploration would encompass the exploration of more parameters and their interrelations. Despite its limitations the given example hints at a more general property of emergent phenomena. It appears that emergent phenomena might require a certain amount of *local* information, i.e. a local context of a certain size that is neither too small nor too big.

Observations such as these, gathered from modeling and simulation of different complex systems within the same framework, may lead to a more general understanding of phenomena like self-organization and emergence.

7 CONCLUSIONS AND OUTLOOK

In this paper we presented a new model for the description and simulation of complex information processing systems. We designed this model specifically to support the *exploratory modeling* of complex information processing systems. The proposed model differs from common modeling approaches like agent-based models in a number of ways:

- Model elements are composed out of a set of four basic building blocks that represent functional rather than structural characteristics of a particular complex system. This contrasts existing models[6] where the different model elements such as cells, agents or messages are used disjunctively.

[6]For example *NetLogo* (Wilensky, 1999) or *repast* (North et al., 2006).

- The state of the model – including the information that is sent at a particular time – has an explicit representation. This facilitates the creation and application of measures that describe global, dynamic properties of the system enabling the comparison of systems that would otherwise be too different to be compared in a more direct way.
- The model focuses on the interactions between its information processing entities. The building blocks of the model are designed such that the modeling of complex interactions is facilitated. One aspect of this is the ability of messages to exist over arbitrary long time intervals which, e.g., enables modeling of the spatial propagation of a message.

In our future work we will use the presented model to investigate phenomena observed in neuronal information processing systems as they are a prime example of complex systems. Their operation involves a substantial variety of mechanisms, e.g., fast signaling through action potentials, complex integration of signals due to dendrite morphology and electrotonic properties, change of processing characteristics via neuromodulators, or the constant adjustment of network topology by means of synaptic plasticity (Koch, 2004; Shepherd and Grillner, 2010). Furthermore, nervous systems develop by evolutionary processes over millions of years and are able to "bootstrap" themselves by a process of self-organized growth within a timescale of weeks to months (Butler and Hodos, 2005; Squire et al., 2008).

Among the first phenomena that we want to analyse are the mechanisms of self-organization utilized during nervous system development such as biochemical guidance cues which are used in the context of neuron migration and axonal path finding.

We think that modeling and simulation of these neuronal systems will not only result in a better understanding of the particular processes themselves but may provide also insight into more general principles of self-organization and emergence.

REFERENCES

Allan, R. (2010). Survey of agent based modelling and simulation tools. Technical report, STFC Daresbury Laboratory, Computational Science and Engineering Department, Daresbury, Warrington WA4 4AD.

Bankes, S. (1993). Exploratory modeling for policy analysis. *Operations Research*, 41(3):435–449.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287.

Brun, R. and Rademakers, F. (1996). Root - an object oriented data analysis framework. In *AIHENP'96 Workshop, Lausane*, volume 389, pages 81–86.

Butler, A. and Hodos, W. (2005). *Comparative Vertebrate Neuroanatomy: Evolution and Adaptation*. Wiley.

Caron-Lormier, G., Humphry, R., Bohan, D., Hawes, C., and Thorbek, P. (2008). Asynchronous and synchronous updating in individual-based models. *Ecological Modelling*, 212(3-4):522–527.

DeAngelis, D. L. and Mooij, W. M. (2005). Individual-based modeling of eecological and evolutionary processes. *Annual Review of Ecology, Evolution, and Systematics*, 36(1):147–168.

Dennett, D. (1998). *Brainchildren: Essays on Designing Minds, 1984-1996*. Representation and Mind Series. Mit Press.

Evans, K. M. (2001). Larger than life: Digital creatures in a family of two-dimensional cellular automata. In Cori, R., Mazoyer, J., Morvan, M., and Mosseri, R., editors, *Discrete Models: Combinatorics, Computation, and Geometry, DM-CCG 2001*, volume AA of *DMTCS Proceedings*, pages 177–192. Discrete Mathematics and Theoretical Computer Science.

Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'. *j-SCI-AMER*, 223(4):120–123.

Holland, J. (1998). *Emergence*. Oxford University Press, New York.

Koch, C. (2004). *Biophysics of Computation: Information Processing in Single Neurons*. Computational Neuroscience Series. Oxford University Press, USA.

Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(13):120 – 149. Proceedings of the Fifth Annual International Conference.

Lempert, R. J., Schlesinger, M. E., and Bankes, S. C. (1996). When we dont know the costs or the benefits: Adaptive strategies for abating climate change. *Change, Climactic Change*, 33:235–274.

Maslow, A. and Wirth, A. (1966). *The psychology of science: a reconnaissance*, volume 8 of *The John Dewey Society lectureship series*. Harper & Row.

Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The swarm simulation system: A toolkit for building multi-agent simulations. Santa Fe Institute.

Newman, M. (2003). The structure and function of complex networks. *SIAM Review*, 45(2):167–256.

Newman, M. E. J. (2011). Resource letter cs-1: Complex systems. *American Journal of Physics*, 79:800–810.

North, M., Collier, N., and Vos, J. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Trans. Model. Comput. Simul.*, 16(1):1–25.

Railsback, S., Lytinen, S., and Jackson, S. (2006). Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9):609–623.

Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th* *annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA. ACM.

Salamon, T. (2011). *Design of Agent-Based Models : Developing Computer Simulations for a Better Understanding of Social Processes*. Academic series. Bruckner Publishing, Repin, Czech Republic.

Shepherd, G. and Grillner, S. (2010). *Handbook of Brain Microcircuits*. Oxford University Press, USA.

Squire, L., Bloom, F., Spitzer, N., Squire, L., Berg, D., du Lac, S., and Ghosh, A. (2008). *Fundamental Neuroscience*. Fundamental Neuroscience Series. Elsevier Science.

Wilensky, U. (1999). *NetLogo http://ccl.northwestern.edu/netlogo/*. Center for Connected Learning and Computer-Based Modeling, Northwestern University., Evanston, IL.