A Survey of Models and Design Methods for Self-Organizing Networked Systems

Wilfried Elmenreich[1], Raissa D'Souza[2], Christian Bettstetter[1], and Hermann de Meer[3]

[1] University of Klagenfurt and Lakeside Labs, Austria
firstname.*lastname*@uni-klu.ac.at

[2] University of California at Davis and Santa Fe Institute, USA
raissa@cse.ucdavis.edu

[3] University of Passau, Germany
demeer@uni-passau.de

Abstract. Self-organization, whereby through purely local interactions, global order and structure emerge, is studied broadly across many fields of science, economics, and engineering. We review several existing methods and modeling techniques used to understand self-organization in a general manner. We then present implementation concepts and case studies for applying these principles for the design and deployment of robust self-organizing networked systems.

1 Introduction

The term self-organization was introduced by Ashby in the 1940s [1]. He referred to pattern formation occurring by the cooperative behavior of individual entities. Such formation can be described by entities achieving their structure without any external influence. The ideas behind self-organization were subsequently further studied and developed by a number of cyberneticians (e.g., von Foerster, Pask, Beer, and Wiener), chemists (e.g., Prigogine), and physicists (e.g., Haken). In the 1980s and 90s, the field was further fertilized by some applied mathematics disciplines, such as non-linear dynamics, chaos theory, and complex networks. Although there is still no commonly accepted *exact* definition of a self-organizing system that holds across several scientific disciplines, we refer to it as a set of entities that achieves a global system behavior via local interactions between its entities without centralized control [2].

Phenomena of self-organization can be found in many disciplines. A well-known example from nature is the flocking behavior in a school of fish. It is likely that there is no "leader fish," but each individual fish has knowledge only about its neighbors [3]. Despite (or probably because of) this localized and decentralized operation, the difficult task of forming and maintaining a scalable and highly adaptive shoal can be achieved.

With the increasing complexity in technology and its applications (more and more entities are interconnected to form a networked system), the notion of self-organization has also become an interesting paradigm for solving technical problems (see, e.g., references in [4]). In order to design a self-organizing technical system, a set of modeling approaches and design methods are needed. The goal of this paper is to give a survey of building blocks that can be used for modeling and design of self-organization, with a special focus on information and communications and traffic systems.

The paper is structured as follows: In Section 2, we take a closer look at specific modeling issues for self-organizing systems. Section 3 presents an overview on different established models for self-organizing systems. Section 4 reviews different methods for designing a self-organizing system with an intended technical effect. Section 5 give references to some case studies with respect to the concepts described in Sections 3 and 4. Finally, Section 6 concludes the paper.

2 Two Perspectives on Self-Organizing Systems

A self-organizing system (SOS) consists of a set of entities that interact with each other locally to obtain a global system behavior. The global system behavior arises out of many simple interactions. It is an emergent property, i.e., it cannot be explained by summation of the local interactions. A self-organizing system can thus be viewed from two perspectives: the microscopic perspective describes the entities and their behavior; the macroscopic perspective describes the (emergent) behavior of the overall system.

An example for these two perspectives and their interrelation can be found in the "predator-prey system" as described by Lotka and Volterra [5]. This biologically-inspired model contains two types of animals, typically named "rabbits" (the prey) and "foxes" (the predators). It obeys simple microscopic rules:

- Rabbits reproduce at a given birth rate ϵ_1.
- Rabbits die if caught by foxes, with rate for such a death event given by γ_1.
- Foxes reproduce at a rate γ_2 proportional to the number of caught rabbits.
- Foxes die according to a given death rate ϵ_2.

These rules can be described in an aggregated form by the differential equations

$$\frac{dN_1}{dt} = N_1(\epsilon_1 - \gamma_1 N_2), \qquad \frac{dN_2}{dt} = -N_2(\epsilon_2 - \gamma_2 N_1)\,,$$

where N_1 and N_2 is the number of rabbits or foxes, respectively. The rate of interaction between both populations is a function of the product $N_1 N_2$.

When examining the behavior of this system over time, we observe emergent phenomena on the macroscopic level, such as periodic oscillations of N_1 and N_2. Unlike many oscillations in nature which can be described by trigonometric functions, the solution of the Lotka-Volterra equations do not have a simple expression in terms of trigonometric functions (see Fig. 1).

To describe the overall system behavior from the macroscopic perspective, Lotka and Volterra found three laws:

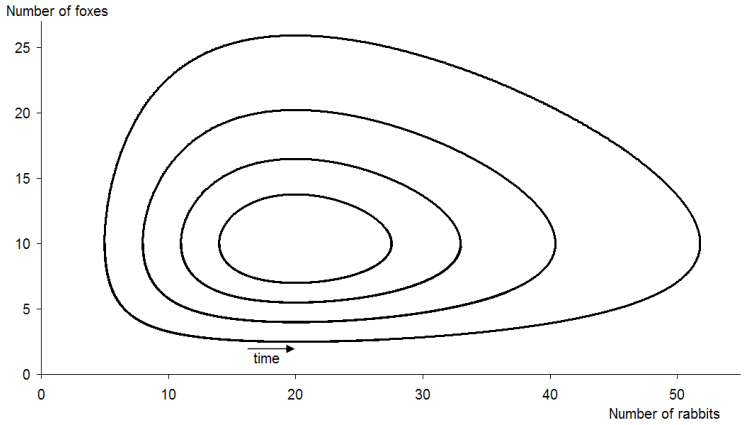

Fig. 1. Simulation with 4 different initial conditions of Lotka-Volterra system with parameters $\epsilon_1 = 1, \gamma_1 = 0.1, \epsilon_2 = 1, \gamma_2 = 0.05$

1. The population sizes of rabbits and foxes oscillate with a particular phase offset. Period lengths depend on the initial population and model parameters.
2. The averages of the two population sizes converge to a value that is determined by the model parameters and is independent of initial conditions.
3. If both populations are decimated proportionally to their size, the rabbit population will recover fast and will exceed their previous population.

These laws describe properties of the overall system, which cannot be immediately seen from the four microscopic rules.

The Lotka-Volterra model enables us to derive both perspectives from the differential equations. For more complex (more realistic) systems, however, the direct transformation between microscopic and macroscopic perspectives might not be possible. In this case, the macroscopic model could be built by (statistical) analysis of the overall behavior. This requires appropriate statistical tools to analyze emergent structures and behavior [6]. Such a model, although being potentially very useful in some cases, is unlikely to cover all possible aspects of the system behavior. An example is the global economy — although tremendous research efforts are put into understanding both its microscopic and macroscopic rules, the system is far from being exactly predictable.

To gain a deep understanding of a self-organizing system or to design a new system, it is beneficial to consider both the microscopic and macroscopic perspective. A microscopic perspective confers the advantage of providing an exact model, which directly supports implementation of the entities. However, it provides no mechanism for measuring or understanding emergent properties. A macroscopic perspective, in contrast, covers emergent behavior and a goal-oriented view. It is less exact, as it treats the system as a "black box." This obscures underlying dynamics and makes it difficult to verify the applicability of the model to a given system.

3 Models for Self-Organizing Systems

Before building and deploying an engineered SOS, it can be beneficial to develop a model of the system which can be analyzed mathematically or explored through computer simulation. This is especially important given that SOS typically display emergent behaviors which, by definition, can not be anticipated in advance. Models provide a virtual laboratory for exploring system design and function and give insights into the types of emergent behaviors that might be expected and the anticipated workings of real systems.

3.1 Differential Equations

Perhaps the first examples of modeling SOS reported in the literature are reaction-diffusion equations which generate patterns similar to those observed in Rayleigh-Bénard convection, viscous fingering, spiral waves in chemical reactions and a range of additional structures such as stripes and tilings [7, 8]. Many of these patterns are also observed in biological systems, with differential equations providing models of organism growth and differentiation [7, 9]. Ref [8] provides a comprehensive review of pattern formation as modeled by reaction-diffusion equations. Most recently, systems of differential equations have been used to model the emergence of synchronization in systems of coupled oscillators ranging from fireflies, to neurons, to Josephson junctions [8, 10, 11]. The Lotka-Volterra model discussed above is another prominent example of this approach.

Despite the broad range of applicability, this approach is thus far limited to describing pattern formation, oscillations and other simple collective phenomena. It does not allow much flexibility for diversity of components and assumes a uniform spatial interaction, neglecting more realistic types of connectivity in real-world systems which are often much better described by networks than uniform spatial fields.

3.2 Cellular Automata

Cellular automata models (CAs) are a simulation method for studying SOS which assume the world is a discrete grid (a lattice) and each site on the grid can be found in one of a set of possible discrete states. CAs have been successfully used to model a range of real physical systems from predator-prey systems, to chemical spiral waves, to hydrodynamics [12–15].

A highly desirable aspect of CAs, absent from differential equations, is that they allow for locality. In other words, sites on the grid interact directly only with neighboring sites, thus, consistent with physical law, there is no "action at a distance" and all signals must propagate along a path of connected neighbors. (Yet, similar to differential equations, this requirement of uniform spatial connectivity does not permit irregular networked patterns of interaction.) Another highly desirable quality of CAs is that they readily exhibit interesting behaviors such as forming a range of patterns similar to those observed with differential equations

and also far more complex dynamics such as moving patterns ("gliders") in Conway's Game of Life [16]. Many CAs are computationally universal, i. e., in theory CAs can be used to implement any computation. A fundamental requirement necessary for many (if not most) CAs to exhibit interesting behaviors is synchronous updating (all grid sites update simultaneously). Though not always realistic, as discussed in Sec. 5.1, there are simple algorithms which do allow collections of objects to synchronize, so it may be possible to engineer a system to fit the CA paradigm.

3.3 Agent-based Models

Agent-based models are another modeling approach and a natural starting point if components in a system can vary dramatically from one another and display a range of behaviors and strategies (especially decision making). In contrast, CAs assume every site in the grid obeys the same rules. Furthermore, agent based models do not need to assume an underlying topology, specifying or restricting which components can interact; agents can come together and interact with one another in complex and dynamic ways. Such approaches are clearly necessary if, as in economics, we wish to model interactions amongst humans with complex decision making abilities. Agent based approaches can also be quite useful even if the agents are simple, such as capturing swarm behaviors [3].

Agent based approaches are extremely flexible, modeling the interactions of a collection of autonomous agents. Thus, unlike CAs and differential equations, they can model phenomena in social systems and agent based approaches lend themselves readily to incorporating game theoretic policies [17]. Many software platforms and techniques exist for building agent based models and Refs. [18] and [19] provide useful reviews.

A major drawback to agent based modeling is the lack of rigor. Due to the complexity of the model specifications (behaviors of agents, patterns of interaction) it is difficult to assess the robustness of observed phenomena to changes in the specifications and the accuracy by which the model describes real systems.

4 Design Methods for Self-Organizing Systems

The design of self-organizing systems differs from typical engineering approaches in the way that the system is rather built bottom-up than top down. At an early stage of development, it is necessary to tinker with the interactions between the individual system entities. In contrast, traditional systems are typically built starting with the overall system service and then approach the micro level only after several more and more fine-grained system models. Therefore, most standard design approaches do not fit well for the design of self-organizing systems. In the following, we discuss several design approaches that have been used or could be used for this purpose.

4.1 Analytical Approach

If the chosen model is abstract and simple enough, the settings for the desired global properties could be derived by an analytical solution. For example, if a system S fed with a configuration C gives the emergent behavior B, the task is to find the inverted function, i.e.,

$$S(C) = B \quad \Rightarrow \quad C = S^{-1}(B)$$

Unfortunately, even for moderately complex systems this is usually not feasible or would require a high effort for solving a mathematical problem which might already be represented by an less realistic abstraction from the actual problem. E. g., for the Lotka-Volterra example there exists is no complete analytical solution for the differential equation system, i.e., the equations have to be reduced or solved numerically.

An analytical approach may, however, help to discover certain aspects of the system. This might be achieved e.g. by assuming certain conditions or parameters. Thus it can help in the initial system design phase to predict certain aspects and in the final phase by verifying system aspects.

4.2 Applying a Reference Design

There exist many examples of self-organization in different domains, such as biology, physics, mathematics, economics. In order to find a working approach for achieving a particular behavior, a reference design from one of these disciplines can provide a major step towards a successful solution. There are two main paradigms for adopting a reference design: top-down and bottom-up.

In the top-down approach, a technical problem is tackled by looking for examples solving an equivalent problem. The found solution and its principles are then analyzed and re-built in a technical application. Examples of top-down approaches are (a) the design of aeroplane wings by observing the gliding flight of birds and (b) the design of turbulence-reducing winglets by analyzing the wingtips of birds [20].

In the bottom-up (or indirect) approach, the working principle of the system is first abstracted from its natural context. This step is done in a basic research effort that is not yet targeted at the specific application. Afterward, the results are used in particular technical applications. Thus, the indirect approach could also be called a "literature-inspired approach". Examples include (a) the concept of artificial neural networks and (b) the concept of ant foraging behavior being applied to mesh network packet routing [21].

4.3 Trial and Error

Another approach is to explore the effect of different interaction rules at the microscopic level on the global system behavior using a trial-and-error method.

The simplest trial-and-error method would be a Monte-Carlo method, where random configurations are created and tested until the global system shows the

intended behavior. Due to the typically high-dimensional search space, however, randomized trial-and-error approaches are very unlikely to succeed within an acceptable time frame.

Alternatively, the trials can be used to learn about the causality of particular configurations and the global system behavior. Thus, after a reasonable number of test configurations, the tester might be able to apply his/her understanding of the emergent processes to find a local rule set with a desirable configuration.

An auxiliary concept to understand the causality between local interactions and global system behavior is introduced by Gershenson [22] as the the notion of *friction*. Friction is a property of interaction between two entities as well as a property of the overall system. This latter friction is to be minimized. By identifying and analyzing points of friction, an engineer can change the local rules towards better system performance. However, this is not straightforward: in several cases a higher friction for a particular entity is beneficial for the overall system.

Additionally, emergent behavior is often counterintuitive to what is expected by most people. Resnick [23] describes a simple simulation of a self-organizing slime mold. Several experts were asked to predict the influence of a specific parameter change on the system. The answer was binary, i. e., there was a 50% random change of guessing the correct answer. Nevertheless, a significant majority of people, including experts on complex and self-organizing systems, guessed the wrong answer.

4.4 Evolutionary Algorithms

Conventional search algorithms can be applied to search for an optimal or sufficiently good set of local rules. However, the search space is typically too large for an exhaustive search. For these cases, evolutionary algorithms and heuristic search algorithms can be a choice. Examples include genetic algorithms, simulated annealing, swarm-based optimization, and the Sintflut algorithm.

Using evolutionary algorithms requires a "testbed" that allows extensive and safe testing at low cost. Usually, such a testbed consists of a simulation of the target system with a model of the environment and the system itself. However, a simulation always implements an abstraction of the real environment, so after the experiments, a real-world validation is required to create trust in the derived solution.

The most prominent example for evolutionary algorithms are genetic algorithms. A genetic algorithm starts with an initial population of candidate solutions for a multidimensional optimization problem. It wishes to quickly find a near-optimal solution. At each generation, the candidates are randomly mutated or combined. The candidates with the best "fitness" establish the population of the next generation. An example where a genetic algorithm is used to design a self-organizing technical system is given in [24]. It was used to find the interaction behavior for a distributed robot soccer team. The behavior was modeled as an artificial neural network to support an implementation of mutation and combination.

4.5 Markov Models and Finite State Machines

Auer, Wüchner and De Meer propose a method to derive local interaction rules by learning from a reference solution [6]. The reference solution can be any algorithm that performs well for the problem. For example, the reference solution might be built as an omniscient system in a simulation. In many cases, it might not be possible to use this solution for a real application because the perfect information cannot be provided to the algorithm or the algorithm might be too complex to be implemented with reasonable response times. However, the omniscient algorithm can be used as an example for teaching its behavior to distributed entities that use only local information and interactions.

The behavior of the reference agent is analyzed using Markovian analysis and then rebuilt in a Finite State Machine (FSM). Thus, the state machine mimics the statistical behavior of the reference agent. The approach relies thus on the possibility that a suitable reference solution is available and that the behavior can be successfully used by an agent with local perception.

In [6], the application of this method is shown by designing an agent for the game theoretic problem "repeated prisoner's dilemma" [25]. In the design process, an agent having perfect knowledge (including the opponent's decision) is created. Then the behavior of this "perfect" agent is analyzed using Causal State Splitting Reconstruction [26], i. e., a method for building recursive hidden Markov models from discrete-valued time series.

The results are then implemented as FSM controling the behavior of a normal (non-omniscient) agent. The resulting behavior was similar to the well-known *tit-for-tat* strategy including *forgiveness*. Tit-for-tat with forgiveness is known to be a highly effective strategy in the repeated prisoner's dilemma.

4.6 Combining the Approaches

The presented design approaches can also be combined. For example, the reference design method may serve as a starting point, where the system designer applies one of the other methods subsequently after choosing a reference model. Another variant of a design process could involve a "bootstrapping" method, where efforts to understand the effect of local interactions are combined with an analysis of the global emergent behavior. Some effects of local rules could be predicted by mathematical analysis of the interaction. For example, in physics, only the gravitational system of two bodies is analytically solved. Still, the results can be applied to understand the movement of more bodies in our solar system, as long as some influences can be neglected. Statistical approaches such as Markov models or evolutionary algorithms can be a further step in the system design. In the combined approach as depicted in Figure 2, insights from the microscopic level are influencing and improving the design at the macroscopic level and vice versa.

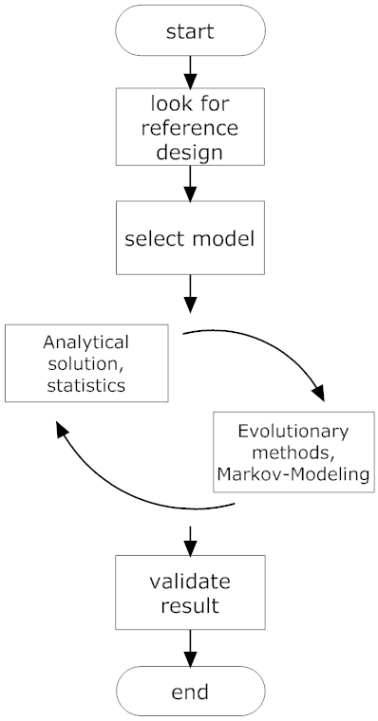

Fig. 2. Combined design approach

5 Case Studies from Engineering

A nice feature of modeling approaches in self-organization is their simplicity at the microscopic level. Simple local rules lead to global structure and function. From an engineering perspective, we would like to apply these models to technical systems. This chapter will give some examples where models from self-organization and complex systems have been successfully applied to information and communications technology research. It will also show that a direct applicability of the theory of self-organization and complex systems is often not possible, but the design of self-organizing functions in technology often requires us to modify and extend the original schemes, taking into account some technological constraints and requirements.

5.1 Wireless Communications: Application of Coupled Oscillators

The pulse-coupled oscillator (PCO) model for firefly synchronization [10] has been employed in many fields of science and engineering. Prominent examples include self-organizing algorithms for time synchronization in wireless systems [27], resource scheduling [28], reducing energy consumption in sensor networks [29], and traffic light control systems [30].

The application of firefly synchronization to mobile and wireless systems requires us to perform some modifications and extensions to the original scheme. These changes are required, since the assumptions in [10] do not match with the system constraints of radio communications. In other words, the modeling assumptions in the original scheme are too simplistic compared to the modeling assumptions typically used in wireless and mobile systems.

First, in general, we cannot neglect inherent delays of the system, including propagation delays, decoding delays, and delays caused by signal processing. These delays make the synchronization scheme unstable, as nodes might receive "echos" of their own firing pulse. To regain stability, a refractory period can be introduced during which nodes do not increase their phase function [31, 27]. Second, wireless communication technologies do typically not allow us to send infinitely short pulses over the air. This fact forces us to replace the infinitely short "firing pulses" by finitely long "synchronization signals" (see e.g. [32, 33]). Third, the wireless medium suffers from noise and attenuation, which must be taken into account for the design of a synchronization scheme as well [34]. Last but not least, to minimize the use of radio resources, it would be beneficial to minimize the signaling overhead, such that nodes only send synchronization words when needed and not periodically as in the original scheme [35].

An approach for self-organized synchronization in wireless systems has been recently developed by Tyrrell, Auer, and Bettstetter [35]. The scheme, called *Meshed Emergent Firefly Synchronization (MEMFIS)*, applies a synchronization word that is common to all nodes in the network and is embedded into each payload data packet. This word is then detected at the receiver using a correlation unit. Starting from an unsynchronized network, synchronization emerges as nodes transmit data packets randomly according to some arrival process. In this way, the throughput of the network can increase gradually, e.g., from ALOHA to Slotted ALOHA.

Another example, where firefly synchronization has been applied to information and security technology is intrusion detection using sensor networks. One approach has been developed in [36].

5.2 Vehicular Traffic: Application of Cellular Automata and Agent-based Approaches

An interesting application of cellular automata, studied extensively in both the physics and the engineering communities, is the modeling and analysis of urban vehicular traffic [37–40]. Gershenson and Rosenblueth [41] apply a two-dimensional model based on simple interaction roles. A street is modeled by a line of connected cells. Each cell can have two states, being empty (0) or occupied by a car (1). The interaction model defines that a car moves on to the next cell in its direction of motion, if this cell is empty; otherwise the car waits. At an intersection, two streets share a common cell. Depending on the state of the traffic lights, this cell operates as a forwarding cell either to the right or downwards, while blocking the other direction, respectively. In their work, Gershenson and Rosenblueth compare a traffic light control algorithm based on a green-wave method and a self-organizing ap-

proach. While the green-wave method requires the cars to match a predefined progression speed to show good throughput, the self-organizing approach shows to be more flexible in adapting to different load situations.

Resnick [23] describes an agent-based traffic model that explains the formation of traffic jams without a centralized cause (such as accidents). Each agent represents a car following a simple set of rules: it slows down if it detects a car close ahead; it speeds up if it does not see a car ahead. In this model, a traffic jam appears as a pattern moving in the opposite direction of the traffic flow. In contrast to cellular automata, the agent-based approach enables a more fine-grained model of the driver's action and decision process. For example, the model could be extended by drivers that have a bad reaction time due to distractions (e.g., phone calls).

6 Conclusions

Several problems in technology and society can be better understood and solved by modeling them as a self-organizing system. An engineer with the task of developing such a self-organizing system faces the problem of modeling and designing the local interactions which will achieve a desired global system behavior. In this paper, we have reviewed several modeling and design approaches suitable in this domain.

Differential equations can model a range of simple collective behaviors, such as pattern formation and the synchronization of coupled oscillators. E. g., the latter are an extremely useful paradigm for modeling self-organizing phenomena; they are often used to describe phenomena related to oscillation and synchronization within a system. Cellular automata are time-discrete and space-discrete models which are often used to display pattern formation phenomena or other phenomena related to the location of the entities. Agent-based models can be applied to both space-continuous and space-discrete phenomena. The are advantageous if entities display a range of behaviors and strategies such as decision making.

The design of a self-organizing system is difficult due to its emergent properties. This paper made an attempt to propose some design approaches, namely the analytic approach, working from a reference design, trial and error, evolutionary algorithms, and a statistic approach based on Markov models. Each of these has certain advantages and disadvantages, thus a combination of them can be useful in the system design process.

Finally, the paper gave examples from communications and traffic engineering where some of the presented models and design approaches have been successfully employed.

Acknowledgments

This paper was supported in part by the KWF (contract KWF 20214/18124/26663 and KWF 20214—18128—26673), the ResumeNet project (EU Framework Programme 7, ICT-2007-2, Grant No. 224619), and the Forschungsrat at the University of Klagenfurt.

This work is an outcome of the Lakeside Research Days 2009 which took place at Lakeside Labs GmbH, Klagenfurt, Austria, from July 13, 2009 to July 17, 2009. The authors would like to thank all participants for the fruitful discussions.

References

1. H. von Foerster. Principles of the self-organizing system. In H. von Foerster and Jr. G. W. Zopf, editors, *Principles of Self-organization*, pages 255–278. Pergamon Press, 1962.
2. W. Elmenreich and H. de Meer. Self-organizing networked systems for technical applications: A discussion on open issues. In J.P.G. Sterbenz. K.A. Hummel, editor, *Proc. Intern. Workshop on Self-Organizing Systems*, pages 1–9. Springer Verlag, 2008.
3. C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proc. Annual Conf. on Computer Graphics and Interactive Techniques SIGGRAPH '87*, pages 25–34, 1987.
4. C. Prehofer and C. Bettstetter. Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine*, pages 78–85, July 2005.
5. V. Volterra. *Leçons sur la théorie mathématique de la lutte pour la Vie*. Gauthier-Villars, Paris, 1931.
6. C. Auer, P. Wüchner, and H. de Meer. A method to derive local interaction strategies for improving cooperation in self-organizing systems. In *Proc. Intern. Workshop on Self-Organizing Systems*, Vienna, Austria, December 2008.
7. A. M. Turing. The chemical basis of morphogenisis. *Philos. Trans. R. Soc. London Ser. B*, 237:37–72, 1952.
8. M. C. Cross and P. C. Hohenberg. Pattern formation outside of equilibrium. *Reviews of Modern Physics*, 65(3):851–1112, 1993.
9. J. D. Murray. *Mathematical Biology: I. An Introduction*. Springer-Verlag, Berlin, 1989.
10. R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, Dec. 1990.
11. A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge University Press, Cambridge, England, 2003.
12. T. Toffoli and N. Margolus. *Cellular automata machines*. The MIT Press, Cambridge, MA, 1986.
13. S. Wolfram. *Theory and applications of cellular automata*. World Scientific, Singapore, 1986.
14. B. Chopard and M. Droz. *Cellular automata modeling of physical systems*. Cambridge University Press, Cambridge, England, 1998.
15. L. B. Kier, P. G. Seybold, and C.-K. Cheng. *Modeling Chemical Systems Using Cellular Automata*. Springer Netherlands, 2005.
16. M. Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, October 1970.
17. R. Axelrod. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press, 1997.
18. N. Gilbert and S. Bankes. Platforms and methods for agent-based modeling. *Proc. Natl. Acad. Sci. U.S.A.*, 99(3):7197–7198, 2002.
19. S. F. Railsback, S. L. Lytinen, and S. K. Jackson. Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9):609–623, 2006.

20. R. Faye, R. Laprete, and M. Winter. Blended winglets. *Aero, Boeing*, (17), January 2002.
21. G. di Caro, F. Ducatelle, and L. M. Gambardella. Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *Springer Lecture Notes in Computer Science*, LNCS 3242:461470, 2004.
22. C. Gershenson. *Design and Control of Self-organizing Systems*. PhD thesis, Vrije Universiteit Brussel, 2007.
23. M. Resnick. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds (Complex Adaptive Systems)*. The MIT Press, 1997.
24. I. Fehérvári and W. Elmenreich. Evolutionary methods in self-organizing system design. In *Proc. Intern. Conf. on Genetic and Evolutionary Methods*, 2009.
25. A. Tucker. *A two-person dilemma*. Stanford University Press, 1950.
26. C. R. Shalizi and K. L. Shalizi. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In M. Chickering and J. Halpern, editors, *Proc. Conf. on Uncertainty in Artificial Intelligence*, pages 504–511, 2004.
27. R. Mathar and J. Mattfeldt. Pulse-coupled decentral synchronization. *SIAM Journal on Applied Mathematics*, 56(4):1094–1106, Aug. 1996.
28. A. Patel, J. Degesys, and R. Nagpal. Desynchronization: Self-organizing algorithms for periodic resource scheduling. In *Proc. Intern. Conf. on Self-Adaptive and Self-Organizing Systems*, July 2007.
29. R. Leidenfrost and W. Elmenreich. Firefly clock synchronization in an 802.15.4 wireless network. *EURASIP Journal on Embedded Systems*, page 17 p., 2009.
30. B. Bayraktaroglu. Traffic light control system and method. United States Patent 4908615, 1990.
31. U. Ernst, K. Pawelzik, and T. Geisel. Synchronization induced by temporal delays in pulse-coupled oscillators. *Phys. Rev. Lett.*, 74(9):1570–1573, Feb. 1995.
32. G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA, November 2005.
33. A. Tyrrell, G. Auer, and C. Bettstetter. Fireflies as role models for synchronization in ad hoc networks. In *Proc. Intern. Conf. on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS)*, Cavalese, Italy, December 2006.
34. Y.-W. Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE J. Select. Areas Commun.*, 23(5):1085–1099, May 2005.
35. A. Tyrrell, G. Auer, and C. Bettstetter. Emergent slot synchronization in wireless networks. *IEEE Transactions on Mobile Computing*, 2010. Under review.
36. Y.W. Hong and A. Scaglione. Distributed change detection in large scale sensor networks through the synchronization of the pulse-coupled oscillators. In *Proc. IEEE Intern. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, 2004.
37. O. Biham, A. A. Middleton, and D. Levine. Self organization and a dynamical transition in traffic flow models. *Phys. Rev. A*, 46:R6124, 1992.
38. T. Nagatani. The physics of traffic jams. *Rep. Prog. Phys.*, 65(9):1331–1386, 2002.
39. D. Helbing and K. Nagel. The physics of traffic and regional development. *Contemporary Physics*, 45:405–426, 2004.
40. O.K. Tonguz, W. Viriyasitavat, and F. Bai. Modeling urban traffic: A cellular automata approach. *IEEE Communications Magazine*, May 2009.
41. C. Gershenson and D. A. Rosenblueth. Modeling self-organizing traffic lights with elementary cellular automata. C3 report 2009.06, Universidad Nacional Autónoma de México, 2009.