Self-Organization in Sensor Networks

Travis C. Collier and Charles Taylor

UCLA Department of Organismic Biology, Ecology, and Evolution
Box 951606, Los Angeles, CA 90095-1606 USA

Abstract

In an effort to better guide research into self-configuring wireless sensor networks, we discuss a technical definition of the term self-organization. We define a self-organizing system as one where a collection of units coordinate with each other to form a system that adapts to achieve a goal more efficiently. We then lay out some conditions that must hold for a system to meet this definition and discuss some examples of self-organizing systems. Finally, we explore some of the ways this definition applies to wireless sensor networks.

Key words: self-organization, sensor networks

1 Introduction

We are currently engaged in a project using self-configuring and adaptive wireless sensor arrays. One of our goals will be to make them self-organizing, by which we mean only the usual, dictionary definition – able to form into a coherent unity or functioning whole. [1] This definition captures the spirit of our intention but is not precise enough for technical application.

Email address: travc@ucla.edu (Travis C. Collier).
URL: http://taylor0.biology.ucla.edu/al (Travis C. Collier).

[1] For example, The Oxford English Dictionary lists several meanings for "organize"; the meaning appropriate to the current context is "2a: to form into a whole with mutually connected and dependent parts; to coordinate parts or elements so as to form a systematic whole (with either the whole or parts as object); to give a definite and orderly structure to; to systematize; to frame and put into working order (an institution, enterprise, etc.); to arrange or 'get up' something involving united action."[1]. To *self*-organize would be to do so autonomously.

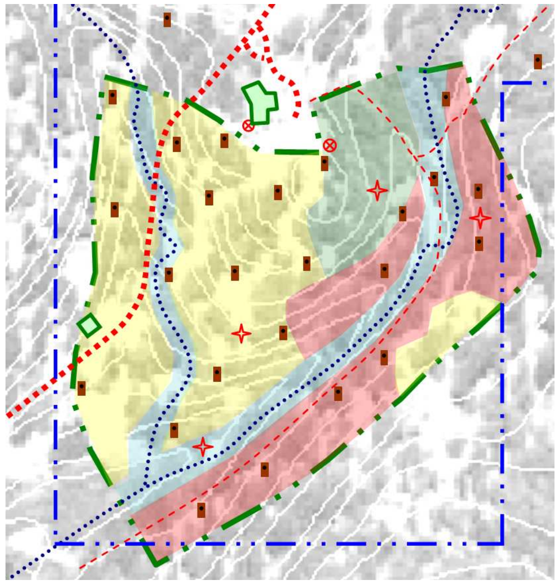

Fig. 1. A portion of the James Reserve sensor array. The figure depicts a 250m wide area colored by vegetation type. Rectangles are nest boxes with sensors, crosses are weather stations, and circles are network cluster heads.

Given the complexity of a sensor network such as the one at the James Reserve depicted in Figure 1, where many sensors of different types are deployed in a heterogeneous and somewhat hostile environment, self-organization is an obvious goal. However, without a more technically suitable definition, it is also a nebulous goal. It would be helpful if we could state clear conditions under which the arrays are capable of self-organizing or not. For example, it might enable us to specify properties that must be engineered in, such as requiring a minimum fidelity of communication among the constituent parts.

"Self-organization" has proven difficult to study scientifically, despite of its centrality to complex systems and adaptive complex systems. On the one hand, most of us share Alan Turing's observation that "Global order can arise from local interactions"[2]. On the other hand, there seems to be no good formulation for the topic that is sufficiently precise[3]. Much has been written of the topic, most of which has been sufficiently abstruse that many scientists "feel a negative visceral reaction to the term 'self-organization' "[4].

Marvin Minsky has distinguished between informal, intuitive use of a word and its appropriateness for technical use[5]. A word may be helpful for general discussion, but when technically precise limiting conditions are required, then the word or concept fails. This does not mean the term is not useful, only that its technical status is limited. Gell-Mann has made a similar point in a discussion about levels of explanation[6]. When pushed to the limit, there seems to be no clear definition of "emergent properties", but the term is nonetheless quite useful for discussing different levels of organization (another concept that is fuzzy and probably not useful for technically precise usage, but still useful for discourse). Possibly "self-organization" is also in this class. We suggest, however, that the term can be made precise, at least in artificial systems designed by humans.

We begin by distinguishing between "self-ordered" and "self-organized" systems and list features necessary for any self-organizing system. We then review some prominent examples of self-organizing systems that have been described and studied in the literature. Finally, we discuss these criteria in more detail as they apply to self-configuring and adaptive wireless sensor arrays.

2 Self-*ordering* vs self-*organizing*

A recent authoritative review[7] defines self-organization as

> "... a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern."

While appealing, it seems to us that this definition pushes back the problem to defining other terms – e.g. levels of organization, emergence, pattern, and local vs. global information. These terms are very difficult to define precisely, perhaps more difficult than the term "self-organization" itself.

Organization involves something more than simply pattern. We prefer to follow Cairns-Smith by distinguishing between systems that are *self-organized* from those that are simply *self-ordered*[8]. In a self-ordered system, the state of part of the system influences the state of another part through local interactions. Examples of self-ordered, but not self-organized, systems would include sand forming waves on the desert floor, water molecules that assemble to form ice, chemical oscillations like the Belousov-Zhabotinskii reaction, molecules that fall into place to form crystals, and arrays of bipolar magnetized needles that settle in to homogeneously-aligned regions. Certain cellular automata might also fit this definition[9,10]. Critical here is the distinction between pattern and function. These examples of self-ordering systems all achieve some order from greater randomness, but there is little that would compel us to regard them as achieving a coherent or functioning whole – essential to the meaning of "organization".

3 Features of self-organizing systems

In our view, a system is self-organizing if a collection of units coordinate with each other to form a system that adapts to achieve a goal more efficiently. Roughly speaking, they form an ensemble, rather than a mere collection. It seems inescapable that the system be viewed as having a goal or goals that the ensemble seems to pursue or in some way optimize. It should be possible to view the system as having inputs, with respect to which it acquires complexity, and some sort of output which it accomplishes more or less efficiently. Adaptation is internalizing information encoded from the inputs to more efficiently or accurately produce the “desired” output.

The importance of the system’s environment should not be underestimated. Inputs, output, and adaptation are all explicitly dependent on the particulars of the environment the system is in. We must expect that in some environments a system might be self-organizing, while in others it might not be.

The role of “goals” in this discussion is problematic. There are deep philosophical issues about whether natural systems can be regarded as having goals and whether one can legitimately infer what these goals are (see [11] for a variety of viewpoints and further references.) We take no stance on that. Rather, we will opt out of that debate and narrow our discussion to *engineered* systems where we can unambiguously designate the goals of the system ourselves.

This definition of self-organization can be enumerated into a list of features.

(1) The system is composed of units which may individually respond to local stimuli.
(2) The units act together to achieve a division of labor.
(3) The overall system adapts to achieve a goal or goals more efficiently.

For these features to hold, the following are some of the conditions that must be met.

(a) The system must have inputs and some measurable output.
(b) The system must have a goal or goals.
(c) The units must change internal state based on their inputs and the states of other units.
(d) No single unit or non-communicative subset of units can achieve the system’s goal as well as the collection can.
(e) As it gains experience in a specified environment, the system achieves its goals more efficiently and/or accurately, on average

The three primary features appear sufficient to classify a human-engineered system as self-organizing. We intend that each of these features will be evident in the discussion of examples to follow.

4 Examples of self-organized systems

Several engineered systems have been described as self-organizing and studied sufficiently well that they can serve as models. These include: (1) self-organizing neural networks; (2) swarm intelligence; (3) self-configuring wireless networks; and (4) cultural acquisition of a common language. This list is by no means inclusive. Important omissions include economic systems[12], metazooan body structure[13], ant colonies[14,7,15], and biological systems in general[16]. We don't dispute that these examples are self-organizing, but because we wish to state precise conditions, we would finesse the question of just what are the goals of these systems. Our discussion will be directed at instances where there are well-analyzed theoretical abstractions of the processes concerned.

4.1 Self-organizing neural networks

Neural networks come in many flavors, some emphasizing biological realism, some designed for engineering efficiency, and others for ease of theoretical analysis. Among the most obvious examples of self-organization are unsupervised learning using Hopfield networks[17], the Willshaw and von der Marlsburg scheme[3], and Kohonen's self-organizing maps (SOM)[18]. There are several excellent treatments of these. For this discussion we will focus on SOM, and more or less follow the treatment of Haykin[19].

In general, SOM are unsupervised learning systems employed to map high dimension inputs to a lower dimension output where similar inputs are mapped near each other. Sophisticated variations have been developed for various applications and mathematical and statistical properties of SOM have been explored [see Kohonen's 1995 monograph[20]].

Imagine a two-dimensional lattice of output neurons all connected to a common input. The input is in the form of an n-dimensional vector, and each neuron contains a single model vector of the same dimensionality. This configuration is illustrated in Figure 2A.

When an input comes into the system, each node computes the distance between that input and its model vector. The node with the closest model vector is chosen to fire either by a master control, a decentralized inhibition scheme, or something in between. Learning is accomplished by the firing node moving

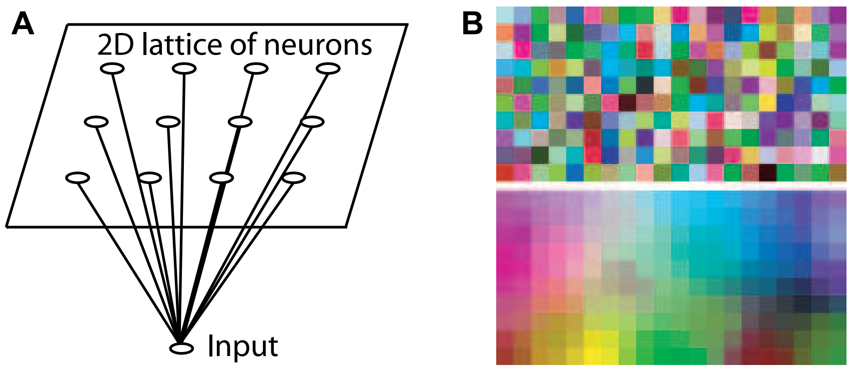

Fig. 2. **A**: The topological configuration of a Self-Organizing Map. **B**: An example of a SOM mapping uniformly random 3-dimensional (RGB color) input vectors to a 2-dimensional map. The top is the initial random configuration, and the bottom is the organized map.

its model vector closer to the last input vector by some delta. The firing node then communicates to each of its neighbors prompting them to do the same with a smaller delta. As those neighbors communicate to their neighbors and so on, a wave of decreasing adjustment toward the last input vector propagates across the network. Typically, the wave of communication is limited to a certain radius centered on the firing node to reduce communication costs, and the delta value decreases with time so that the network is guaranteed to converge.

The output of the system is the identity of the node that fires given a certain input. The two-dimensional lattice in our example is not a real constraint as a huge number of output topologies can be constructed simply by changing how neighbors are defined. Starting from random connections and completely without being informed of what are correct or incorrect responses aside from their own recognition, the system acquiring a *feature map*, Φ, that takes spatially continuous input vectors, $\mathbf{x}$ from a space $\mathcal{H}$ and maps them to a discrete space, $\mathcal{A}$, the topology of which is determined by the spatial array of the output neurons. Figure 2B shows an example of such a mapping, taking uniformly drawn random RGB color values (3-dimensional) to a discrete 20 by 10 2-D plane.

The coordination of the nodes and division of labor is quite clear in this system. Communication is explicit, as is the response of the internal state in response to the inputs and the reinforcement from neighbors. Division of labor is accomplished by decomposing the memory required to store patterns in the inputs as well as the computation of distance between input and model vectors across the nodes. Also, no noncommunicatve subset of nodes can accomplish the mapping as well as the entire network since the output of the network is implicitly defined by the communicative topology of the nodes.

System-wide adaptation is achieved by creating a map. A precise mathematical formulation of the goal is beyond our scope here, but it intuitively involves creating a map where the distance in the output topology is correlated with the distance observed between the inputs. In terms of the formalization above, the system is able to generalize by mapping a large number of inputs drawn from $\mathcal{H}$ to a smaller region in $\mathcal{A}$. For simple topologies and distance measures, convergence has been shown[21]. Inputs, outputs, a goal, and adaptation toward that goal are all satisfied. By all of the criteria discussed above these systems are self-organizing.

While this system is most often implemented as a single computer program, it is conceptually composed of multiple units, the nodes, that individually respond to the inputs. When a centralized method is used to choose which node fires, it becomes less clear that the nodes themselves take actions; however, by formulating the choice process as decentralized the "collection of agents" view is clear. This raises the important point that it is often possible to describe a system as self-organizing, and then turn around and describe the same system in a way that is not obviously self-organizing. How the system is conceptualized, either as monolithic or as a collection of interacting parts, is sometimes important.

4.2 Swarm intelligence

Swarm intelligence is a technique for designing optimization algorithms modeled after the apparent self-organization of social insects. There is a great deal of literature on these sorts of algorithms as well as dedicated conferences. For our discussion, we will roughly follow the brief survey by Bonabeau et al.[22]

One swarm intelligence application, known as Ant Colony Routing, is inspired by the way ants of species such as *Lasius niger* converge on a path when foraging. When an ant forages, it marks its path by laying a pheromone trail. If it returns to the nest with food, more workers will leave the nest to gather food by following pheromone trail reinforcing it as they go. This has inspired an algorithm to solve the Traveling Salesman Problem, or TSP, which involves finding the shortest path between n cities, visiting each only once. Each path between two cities is marked by a pheromone level. A collection of computational agents select routes that visit each city only once, called tours, by choosing between routes probabilistically weighted by the pheromone level on each segment. Then each agent lays down more pheromone on each route in its tour proportional to how good (or short) the overall tour is. By iterating this process and having the pheromone trails decay over time, the agents discover a tour that is quite short if not always the absolute shortest. With slight modifications, this algorithm performs very competitively compared with all other state of the art algorithms.

The same notion of agents probabilistically following and reinforcing a virtual pheromone trail has led to a host of other algorithms in scheduling and routing. Network routing using the ant colony metaphor, known as Ant Colony Routing, is particularly noteworthy as it often produces better results than other commonly used standards and is used commercially[23].

While these algorithms could be formulated in terms of reinforcement optimization, the explicit formulation in terms of computational agents is useful in designing them. As with SOM, one could describe them as a monolithic implementation that is not self-organizing by our definition, but describing the algorithms in terms of a collection of agents is part of the point of using the ant colony metaphor in the first place. In contrast to SOM, agents in swarm intelligence systems keep very little internal state and do not communicate with each other directly. Their state is typically just their location and communication is achieved by depositing the virtual pheromone trail. This does satisfy our criteria for self organization, as collaboration between the agents is achieved by their collective action of depositing pheromones. Part of short tours in the TSP, or uncongested links in the routing algorithm, are strongly marked since they are used by many agents.

The inputs and outputs of the system as a whole also contrast SOM. The input in the case of the TSP algorithm is the static topology of the cities, and the output is the best observed tour, list of cities in order, an agent makes. The goal is explicit, finding the shortest tour, and has been shown empirically to be optimized by the system as a whole[24]. For ant colony routing, the input is again the topology of the network, but the topology changes as traffic patterns change and nodes are added or fail. The goal which is optimized by the system is an implicit function of throughput and latency of traffic coming into the system, maximizing the former and minimizing the latter.

Swarm intelligence is a paradigm for producing efficient and adaptive agent based algorithms. Unsurprisingly, it provides many good examples of self-organizing systems, of which routing is but one.

4.3 Self-configuring wireless networks

The challenge of producing a well connected network on small wireless devices leads to some of the clearest examples of self-organization available. Limited power, low radio range, potentially high density, and an ever changing environment all push for a distributed and adaptive solution. These are some of the same challenges facing wireless sensor network design[25].

One problem faced by wireless sensor networks is how multiple units can coordinate their communication, since no two units in the same area can successfully transmit on the same frequency at the same time. This is the "media access control problem" and has generated huge amounts of study, including relatively early talk of self-organization[26]. One approach is to assign each unit a specific time slot when they are allowed to communicate. Schemes along this line, called TDMA for "Time Division Multiple Access", work reasonably well, but are inefficient when there are many units that are not all in range with each other. They also require that all of the units to have precisely synchronized clocks, which requires costly global communication.

An alternative scheme is for each unit to coordinate only with its neighbors, the units its radio can directly reach. This notion has been taken to quite a sophisticated level including a system called S-MAC specifically designed for sensor networks[27]. Among other things, S-MAC has local groups of nodes coordinate "sleep schedules" requiring them to turn on their radios for only short periods of time to conserve energy.

Beyond initially determining media access control and routing information, wireless networks must continuously deal with the connectivity topology changing. Units may fail, units, or their surroundings, move around, and the weather might even shift and change which nodes are within radio range of each other. Self-organization through continuous local coordination of the units is often the design paradigm of choice to handle these complications.

Unlike the SOM and swarm intelligence examples here, wireless networks are physically composed of communicating units. Collaboration is necessary between the units not only for medium access control, but also time synchronization and routing. On the other hand, the goal is implicit. Just as with swarm intelligence routing, the goal of a wireless network is to provide a high throughput, low latency connection between any two units in the network. This may be modified, especially in the case of a sensor network, to providing a connection between data sources and data sinks, which are not necessarily uniformly distributed across the network. That is, some units may produce traffic more often, and others may be the destination of that traffic. High throughput between those points is more important than high throughput to normally quiet units.

4.4 Cultural acquisition of a common language

Acquisition of a common language by a collection of units is a less well explored self-organizing system. One instantiation of such a system is a network of sensors developing a high-level language among themselves to communicate observed events. Among the motivations for the use of such a learned language include the ability to communicate using very short transmissions as opposed to sending the raw data of observations and the ability to express observations that were not anticipated by the designer of the system.

The mechanisms of event perception and perceptual grounding of utterances provide the basic ingredients of the development of language in a population. This idea has been explored by several researchers with promising but still quite limited success.

Steels explored how shared simple meanings might emerge from simple language games[28,29], ignoring syntax. Hashimoto and Ikegami[30] have explored a more mathematical perspective by creating an evolutionary language game.

Kirby has constructed systems that can evolve the ability to tie observations to statements having a well-defined syntax[31]. He assumes that meanings are given (avoiding the event perception problem) and he represents them in a standard predicate-argument notation, e.g. `kills(stan,kenny)`. This *meaning* is drawn from a semantic space of atomic concepts — simple nouns and verbs such as"john", "alice", "loves", "kills", "admires", or "believes" — that are combined into simple predicate-object propositions, such as `admires(john,alice)` or `believes(mary,admires(john,alice))`. Note that there can be limitless nesting, e.g. with "believes". There are *utterances* that consist of strings of alphabetic characters which are associated with meanings, such as <`alicelovesjohn, loves(alice,john)`> or <`gjtejfqpb, admires(mary,alice)`>. Finally, there are *grammars*, simple definite—clause grammars, such as:

$$\text{S/loves(alice,john)} \rightarrow \text{alicelovesjohn} \left\| \begin{array}{l} \text{S/p(x,y)} \rightarrow \text{N/x V/p N/y} \\ \text{V/loves} \rightarrow \text{loves} \\ \text{N/alice} \rightarrow \text{alice} \\ \text{N/john} \rightarrow \text{john} \end{array} \right.$$

One agent (the speaker) sends an utterance/meaning pair to another (the learner), e.g. <`alicelovesjohn, loves(alice,john)`>, and the pair is added to the learner's grammar. The utterance/meaning pair defines a rule for the grammar. After each rule is incorporated there is a search for generalizations, e.g.

$$\text{S/loves(alice, john)} \rightarrow \text{alicelovesjohn}$$

$$\text{S/loves(alice, george)} \rightarrow \text{alicelovesgeorge}$$

gets compressed to

$$\text{S/loves(alice, x)} \rightarrow \text{aliceloves N/x}$$

and the following rules get added to the grammar:

$$\text{N/x} \rightarrow \text{john}$$

$$\text{N/x} \rightarrow \text{george.}$$

There is a facility for invention, so if an agent does not know how to say what it wants, then it makes up a string at random, and adds that to its grammar as well as stating it.

Clearly there is a lot in the combining algorithm. An outline of the induction algorithm is given in [31], and it is spelled out more fully in [32]. One speaker presents a fixed number of utterance/meaning pairs to a learner. The speaker is then removed. The learner becomes the speaker, and a new agent becomes the learner. This is repeated many times. Kirby observed that languages evolved from random utterances: strings became paired with atomic concepts and grammars evolved that composed these.

When this sort of language game is applied to a large collection of agents, all members of the population come to share a common meaning when they hear a similar sentence. The agents may converge to a common understanding of utterances under certain conditions. In terms of self-organization, the system adapts to develop a mutually understandable language to express the perceptual events which are the inputs into the system. The output of the system is the shared perception-to-meaning mapping in the form of a grammar.

5 Self-organization of sensor arrays

As noted in the discussion of wireless networks, wireless sensor arrays are a good example of a self-organizing system in the way they form a coherent network. However, in sensor arrays we have the opportunity to make them self-organizing in an even more interesting way by having them perform data fusion. Pottie and Kaiser describe the goal of a sensor network as "given a set of observables $\{X_j\}$ to determine which of several hypothesis $\{h_i\}$ are true"[25]. This is accomplished by extracting a feature set from the observables and choosing the hypothesis that has the highest probability given that feature set. This deviates from basic decision theory because each node in the sensor network has only some of the observables and therefore cannot make the best decision, yet communicating all of the observed data to a central node is typically far too costly given constraints on energy. Combining useful information from multiple nodes, called data fusion, often allows for a better system overall.

The primary challenge of data fusion is for the sensor nodes to extract and transmit only the *useful* information[33]. Imagine a dense cluster of seismic sensors deployed to detect and localize traffic. A vehicle enters the field of sensors, providing input to many, though perhaps not all, nodes in one area. The nodes transmit to each other that they have spotted something. By combining just the timing information from two nodes, a directional time-of-arrival computation can be performed to give some location information. The location can more accurately be determined by involving more nodes in the calculation, but nodes that are very near the first two have little they can contribute to localization. What is needed is for the nodes to be able to predict how much they can contribute to the calculation, and only transmit if they can provide a significant improvement. There are algorithms that can provide this particular form of self-organization[34,35].

The network need not pass its raw data after localization has occurred, but instead just the summary "at time T, something was spotted at location X." Similarly, data fusion techniques could potentially have the nodes nearest the vehicle analyze their data and come to a classification such as "truck", "tank", or "auto". Multiple modalities of sensing can also be fused to provide an even greater gain.

Consider how nodes in a sensor array might self-organize to generate an adaptive language they can communicate with. This is a self-organizing coding scheme in a sense, since it provides the ability to communicate a great deal of information with few bits. An adaptive language might be particularly useful when we consider a heterogeneous sensor network, where nodes have different sensing modalities from each other. In this case, a single event may be viewed very differently between, say, a seismic and a magnetic sensor. For data fusion, it is important that the nodes be able to recognize that they are sensing the same event, and therefore need language to communicate the "meaning" of their observations that is somewhat independent of their sensor modality.

Beyond language development, sensor arrays can also employ distributed storage, hierarchical structure, and potentially mobility, all of which lead to interesting self-organizing implementations.

What criteria should be used to define whether or not a sensor array is self-organizing? Consider an example: a field of distributed sensors, on the order of 10 - 100, distributed throughout a field. Each sensor can be one of several types, in this case three – seismic sensor, acoustic sensor, or designated wireless communicator to the central human processor. For each sensor there will be a state-dependent cost of sensing, cost of processing, and cost of communicating. The task of the sensor array is to localize a source moving through the field, and one would like to maximize accuracy, life of the system and stealth (which would be related to communication expenditures).

The *goal* of this system is to follow a single source, minimizing the tracking error (integrated over time), while at the same time minimizing the power consumption and communication. We define the system as *organized*, with respect to the goal just described, if it minimizes the the weighted sum of *tracking error*, *power consumption*, and *communication*.

Organization is reduced to an optimization problem by this approach. We can then define *self-organization* as a constraint on the method whereby the system becomes organized autonomously.

6 Discussion

We have attempted to be precise and concrete at the expense of being general. For example, a colony of leafcutter ants, *Atta* sp., surely exemplifies self-organization in the colloquial sense[15]. Because of problems with identifying goals, however, they would not qualify as self-organizing under the definition proposed here. It might be possible to say that this ant behavior is self-organizing *with respect to fitness* or *with respect to food-gathering*. This suggests that we could assign a goal to a natural system, rather than simply infer one, then speak of self-organization with respect to that assigned goal. Note that this approach would lead to inferring self-organization with respect to some goals, but not necessarily to others.

Goals in our sense refers to the ensemble – and each individual might not share that goal. We observed earlier that some, perhaps all, self-organizing systems can be conceived either as distributed and self-organizing, or as a single system about which self-organization makes little sense. This is related to the fact that all distributed processes can be simulated by a single process (though not all single processes can be simulated by distributed ones). But it is not so simple. There are several features of self-organizing, multi-agent systems that are not typically shared by monolithic systems. These include the ability to recover from damage and a distributed presence. While a network of sensors that loses half of its members to being crushed might still be able to process in a self-organized manner, albeit less efficiently, a single processor, half of which is crushed, is unlikely to recover (see, e.g. Twain's story of Puddin'head Wilson[36]). Clearly, how a system is conceived of logically is only half the story.

Finally, the utility of this approach will be determined largely by the value of those generalizations that can be made about self-organizing systems. One of the earliest mathematical studies of self-organization was the monograph by Eigen and Schuster on hypercycles[37]. They found that certain conditions on fidelity of reproduction are required for hypercycles to grow or even to persist. Komarova and Nowak found that these same conditions, more or less, apply to the fidelity of word usage in self-organizing languages[38]. Resnick made a related observation noting that ants that lay pheromone trails to food can only maintain those trails if the density of ants following the trail is sufficiently high[39]. These exemplify the generalization that a minimum combination of density and communication is necessary for information to percolate through a system and allow it to be self-organizing. Probably the same is true for the density and fidelity of communication in sensor networks.

Will there be enough other generalities about self-organizing systems to make this approach useful? Von der Marlsburg offered some heuristics for self-organization, but observed that "A canonical mathematical formulation of self-organization has yet to be developed"[3]. If our approach is to rise above a "distinction without a difference" then it must lead to other generalizations that are found useful. Other generalizations are not difficult to identify. The point here is not to enumerate all the possible statements that can be proven about such system, but simply to point out that once a clear definition has been adopted, then (and only then) it is possible to make correspondingly clear statements about the conditions for self-organization to occur.

We expect that engineered self-organizing systems will become more common and important as computation becomes more ubiquitous, as computational explanations become more accepted by the general scientific community, and as principles of self-organization become better understood. We hope that the formulation proposed in this paper will allow more precise identification of such systems, allow more general principles about them to be derived, and provide insight into how such systems might more readily be constructed.

Acknowledgments

We would like to thank Lieven Vandenberghe, A. Graham Cairns-Smith, Mike Hamilton of the James Reserve, and the Adaptive Language group at UCLA led by Ed Stabler.

This work was supported by the UCLA Center for Embedded Network Sensors, the Defense Advance Research Projects Agency (DARPA), administered by the Army Research Office under Emergent Surveillance Plexus MURI Award No. DAAD19-01-1-0504, and DARPA MURI award administered by the US Airforce No. F49620-01-1-0361. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and

do not necessarily reflect the views of the sponsoring agencies.

References

[1] J. A. H. Murray, H. Bradley, W. A. Craigie, C. T. Onions (Eds.), The Oxford English Dictionary, 2nd Edition, Clarendon Press, Oxford, 1993.

[2] A. M. Turing, The chemical basis of morphogenesis, Philos. Trans. Roy. Soc. Lond. B 237 (1952) 37–72.

[3] C. von der Marlsburg, Network self-organization, in: S. Zornetzer, J. L. Davis, C. Lau (Eds.), An Introduction to Neural and Electronic Networks, 2nd Edition, Academic Press, San Diego, CA, 1995, pp. 421–432.

[4] D. Jefferson, Personal communication (Aug 1999).

[5] M. L. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, 1967.

[6] M. Gell-Mann, Simplicity and complexity in the description of nature, Engineering and Science LI (3) (1988) 3–9, california Institute of Technology.

[7] S. F. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, Self-Organization in Biological Systems, Princeton University Press, Princeton, NJ, 2001.

[8] G. Cairns-Smith, Personal communication (Aug 2001).

[9] S. Wolfram, Cellular automata as models of complexity, Nature 311 (1984) 419–428.

[10] S. Wolfram, A New Kind of Scienece, Wolfram Media, Champaign, IL, 2002. URL `http://www.wolframscience.com/`

[11] C. Allen, M. Bekoff (Eds.), Nature's Purposes: Analyses of Function and Design in Biology, MIT Press, Cambridge, MA, 1998.

[12] P. R. Krugman, The Self-Organizing Economy, Blackwell Publishers, 1996.

[13] L. W. Buss, The Evolution of Individuality, Princeton University Press, New Jersey, 1987.

[14] D. Gordon, Ants at Work: How An Insect Society Is Organized, Norton, W. W. and Co. Inc., 2000.

[15] B. Hölldobler, E. O. Wilson, The Ants, Belknap Press of Harvard University Press, Cambridge, Massachusetts, 1990.

[16] S. A. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, New York, NY, 1993.

[17] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, PNAS 79 (1982) 2554–2558.

[18] T. Kohonen, Self-organizing formation of topologically correct feature maps, Biol. Cyb. 43 (1) (1982) 59–69.

[19] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 1999.

[20] T. Kohonen, Self-Organizing Maps, Vol. 30 of Springer Series in Information Sciences, Springer, Berlin, Heidelberg, 1995, (Second Extended Edition 1997).

[21] E. Erwin, K. Obermeyer, K. Schulten, Convergence properties of self-organizing maps, in: T. Kohonen, K. Mäkisara, O. Simula, J. Kangas (Eds.), Artificial Neural Networks, Elsevier, Amsterdam, Netherlands, 1991, pp. 409–414.

[22] E. Bonabeau, M. Dorigo, G. Theraulaz, Inspiration from optimization from social insect behaviour, Nature 406 (2000) 39–42.

[23] G. D. Caro, M. Dorigo, Antnet: Distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research 9 (1998) 317–365.
URL `citeseer.nj.nec.com/dicaro98antnet.html`

[24] M. Dorigo, L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.
URL `citeseer.nj.nec.com/article/dorigo97ant.html`

[25] G. J. Pottie, W. J. Kaiser, Wireless integrated network sensors, Comm. of the ACM 43 (5) (2000) 51–58.

[26] T. Robertazzi, P. Sardchik, Self organizing communication networks, IEEE Communications Magazine 24 (1) (1986) 28–33.

[27] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, NY, 2002.

[28] L. Steels, Perceptually grounded meaning creation, in: V. Lesser (Ed.), Proceedings of the First International Conference on Multi–Agent Systems, MIT Press, 1996.
URL `citeseer.nj.nec.com/steels96perceptually.html`

[29] L. Steels, P. Vogt, Grounding adaptive language games in robotic agents, in: C. Husbands, I. Harvey (Eds.), Proceedings of the Fourth European Conference on Artificial Life (ECAL97), MIT Press, 1997.
URL `citeseer.nj.nec.com/steels97grounding.html`

[30] T. Hashimoto, T. Ikegami, Emergence of net-grammar in communicating agents, BioSystems 38 (1996) 1–14.

[31] S. Kirby, Syntax out of learning: the cultural evolution of structured communication in a population of induction algorithms, in: D. Floreano, J. D.

Nicoud, F. Mondada (Eds.), Proceedings of the Fifth European Conference on Artificial Life (ECAL99), 1999, pp. 694–703.
URL `http://ling.ed.ac.uk/anonftp/pub/staff/kirby/ecal99.ps.gz`

[32] S. Kirby, Learning, Bottlenecks and the Evolution of Recursive Syntax, Cambridge University Press, 2002, Ch. 6.
URL `http://ling.ed.ac.uk/anonftp/pub/staff/kirby/ted.ps.gz`

[33] J. M. Bernardo, A. F. M. Smith, Bayesian theory, Wiley, New York, 1996.

[34] H. Wang, K. Yao, G. Pottie, D. Estrin, Entropy-based sensor selection in localization, Tech. Rep. 0011, Center for Embedded Network Sensors (April 2003).

[35] J. Chen, K. Yao, R. Hudson, Source localization and beamforming, IEEE Signal Processing Magazine 19 (2002) 30–39.

[36] M. Twain, Puddin'head Wilson and Those Extraordinary Twins, Penquin, 1894.

[37] M. Eigen, P. Schuster, The Hypercycle:A Principle of Natural Self-Organization, Springer Verlag, Berlin, 1979.

[38] N. L. Komarova, M. A. Nowak, The evolutionary dynamics of the lexical matrix, Bulletin of Mathematical Biology 63 (2001) 451–484.

[39] M. Resnick, Turtles, termites, and traffic jams: explorations in massively parallel microworlds, MIT Press, 1994.