Lints, T. (2010). *Research Laboratory for Proactive Technologies, Department of Computer Control, Tallinn University of Technology, Annual Report 2009*, chapter Self-Organization, pages 24–34.

T. Lints, *Research Laboratory for Proactive Technologies, Department of Computer Control, Tallinn University of Technology, Annual Report 2009*, ch. Self-Organization, pp. 24–34. 2010.

```
@inbook{Lints10,
  author = {Taivo Lints},
  title = {Research Laboratory for Proactive Technologies,
Department of Computer Control, Tallinn University of
Technology, Annual Report 2009},
  year = {2010},
  chapter = {Self-Organization},
  pages = {24-34}
}
```

Self-Organization

Self-organization, if described in a general and intuitive way, is "the appearance of structure or pattern without an external agent imposing it" (Heylighen, 2003), where the pattern should be understood in the broader sense, including patterns of behavior and interactions. It is worth noting that this description does *not* totally prohibit all external interactions, and in real systems the existence of those is often even *required* for the self-organizing processes to function. It is just that the structures and patterns should not be *fully* imprinted / enforced from outside.

The reason why self-organization has become a process of great interest to many fields of science and engineering is, from the analysis side, the hope to use it for explaining a lot of otherwise puzzling behaviors occurring in natural and artificial systems, and from the management and synthesis side the appeal of the idea that it is possible to create complex resilient and highly scalable systems from relatively simple components and rules.

The Development of the Concept

According to Shalizi (2009), the first articulation of the concept of self-organization is likely to be by René Descartes from the first half of the 17th century, and even though there surely also existed earlier theories that did *not* require intelligent design either, they were mostly of the type “given enough time and space and matter, organization is bound to happen somewhere, sometime, by sheer chance”, which is not really about self-organization. Descartes, on the other hand, not only mentioned the concept but also put forward a rather lengthy discussion where it was presented as a hypothetical arrangement the God would have made if He had not wanted to design and develop everything Himself. Basically, the idea that Descartes introduced was that the ordinary laws of nature tend to produce organization. But the times were not supportive for theories that diminish the role of God.

In the 18th century, the term "self-organizing" was used in German philosophy, notably by Immanuel Kant, who argued that *teleology* is a meaningful concept only with regard to systems that are capable of governing itself – systems consisting of parts that exist thanks to, and for the sake of, each other (Wikipedia, 2010). The time period from the beginning of the 19th century to the Second World War saw sporadic mentions of the term, but in addition there was also some quite rigorous research done on specific self-organizing systems without explicitly analyzing self-organization as such. This included the works of William C. Bray on oscillating chemical reactions and Alfred J. Lotka and Vito Volterra on predator-prey dynamics (Karsenti, 2008).

The main interest in self-organization was sparked after World War II when the fields of cybernetics and computing machinery started exploring it as a useful separate theoretical concept on its own. One of the most notable contributions on the topic was from William Ross Ashby. He viewed the *organization* of a system as a restriction or constraint on the space of possibilities, as opposed to something "extra" added to the elementary variables, and noted that different observers might consider different spaces of possibilities, and differently divide the system into parts (Ashby, 1962). He also explained that there is no such thing as "good organization" in any absolute sense, only *relative* to some environment and to some sets of threats, disturbances, problems and goals. And then he provided two general meanings of *self*-organization: 1. Self-change from unorganized to organized, i.e., from parts separate to parts connected; 2. Self-change from a bad organization to a good one, or, from one organization to another. More formally, as described by Shalizi (2001), Ashby defined the *organization* of a system as the functional dependence of its future state on its present state and its current external inputs, if any. That is, given state space **S** and input space **I**, the organization of the system is the function **f**: $\mathbf{S} \times \mathbf{I} \rightarrow \mathbf{S}$ which gives the new state. *Self*-organizing would then mean changing **f** internally, i.e., **f** would be a function of the state, which, Ashby says, would make nonsense of the whole concept (at least within the specific formal framework that he was using). There was a workaround, however, as, again, summarized by Shalizi (2001): **f** might be approximated well by a function **g** in a certain region of the state space and by **h** in another region, and if the dynamics then drive the system from the first region to the second, we see an apparent change in the organization, from **g** to **h**, even though the true, underlying dynamics remain the same.

As Shalizi (2001) notes, despite a few shortcomings – not properly distinguishing changes that lead to more organization from those which lead to less, and not going into as much details mathematically as would be desired – Ashby's work on self-organization was an outstanding contribution, but unfortunately those ideas were pretty thoroughly ignored by everyone else who used the idea in the 20th century. Apart from that ignorance, the research on self-organization did flourish in the second half of the century. To give a few examples: in physics and chemistry there was a lot of self-organization-linked work on pattern formation and spontaneous symmetry breaking (Turing, Belousov, Zhabotinsky, Nicolis, Prigogine), on laser and other cooperative phenomena (Haken), on self-organized criticality (Bak, Tang, Weisenfield, Jensen), in computer science and related fields on machine learning (Selfridge, Kohonen), on adaptation (Holland, Farmer), on distributed computation (Wolfram, Resnick, Crutchfield, M. Mitchell), in economics (Schelling, Krugman), and so forth. Additionally, the interdisciplinary effort on understanding complex systems, under the name *complexity science*, paid self-organization a considerable amount of attention.

Looking at the more recent definitions, the one provided by Camazine et al. (2001) in the context of biology is a rather good representative of the current mainstream viewpoint (especially that of the applications-oriented researchers). They consider self-organization to be a process in which a pattern at the global level of the system emerges solely from numerous interactions among the lower-level components. Moreover, the rules specifying interactions among the system's components should be executed using only local information, without any reference to the global pattern. To make the concept clearer, Camazine et al. contrast it with those organizational processes that have leaders who are giving detailed instructions to the group members, or have blueprints that specify almost all aspects of the organization, or recipes in the sense of precise sequential instructions with nearly no feedback, or templates – full-size guides or molds that strongly steer pattern formation. This definition might be slightly too strict, though, as it tries to *fully* exclude all external organizational influences, whereas in practical systems total isolation is a rare exception. Or, alternatively, and with some reservations, it could be seen as a description of the *pure* theoretical concept of self-organization that in real systems occurs mostly in a diluted and mixed form.

As of the more rigorous recent theoretical works, one worth pointing out is that of Cosma R. Shalizi from the turn of the century that will be shortly and superficially described in the following, based on Shalizi (2001, 2004, and 2009). First he notes that the rather widespread formalization of the increase of organization as the decrease of entropy is somewhat inadequate – (thermodynamic) entropy is a useful measure of order in statistical mechanics, but in a more general context it, among other problems, fails to distinguish between the many different *kinds* of organization systems can exhibit. To illustrate this, Shalizi quotes Peter B. Medawar (1982) that, fundamentally, “biological order is not, or not merely, unmixedupness”.

Shalizi then proposes that a good way to formalize the increase of organization is to equate it with the increase of complexity. More specifically, the measure of complexity used for this formalization should be of the kind that has the maximum in the "interesting" region *between* disorder and order, i.e., a fully regular pattern would *not* be considered complex, and neither would a fully random one. A concept matching this requirement is *statistical complexity*, which measures the minimal amount of information needed to predict the system's behavior, that is, how much information about the past of the system is *relevant* to predicting its future dynamics. The higher the complexity of a system is, the greater the (effective) number of parts in its "causal architecture", which might possibly be represented as a state machine (Shalizi uses *epsilon*-machines that are supposed to be the simplest possible representations of a process that can be inferred from a given stream of data produced by that process). The statistical complexity is then simply the log of the effective number of causal states (effective in the sense that if there exist separate states in the model that are equivalent from the viewpoint of prediction, i.e., lead to the same outcome, they are bundled together into an equivalence class; *epsilon*-machines have this property already by construction). By equating the level of organization of a system with its statistical complexity, we can say that the system has *organized* between times $\mathbf{t_1}$ and $\mathbf{t_2}$ if $\mathbf{C(t_2) - C(t_1) \equiv \Delta C > 0}$, and it has *self*-organized if *some* of that rise in complexity is *not* due to external agents.

Additionally, Shalizi explains that, at least within this framework, self-organization does *not* imply emergence, and vice versa, even though from the practical viewpoint there indeed is a subtle link between them, and neither does self-organization imply adaptation. The latter, though not further elaborated by Shalizi, is quite clear. Adaptation can be talked about only with regard to specific goals, and it is always possible to define some goals that a given self-organizational process supports (thus being adaptational with regard to them), some goals it conflicts with (thus being the opposite of adaptational) and some goals it is irrelevant to. As of the relationship between self-organization and emergence, it is first necessary to specify how emergence is defined. Shalizi considers a (higher level, derived) process to be emergent if it has a greater predictive efficiency than the (lower level, underlying) process it derives from. He then explains why so many authors have linked self-organization and emergence (Shalizi, 2001, page 118):

> When something self-organizes, it becomes more statistically complex, i.e., optimal prediction requires more information. A cognitively-limited observer (such as a human scientist) is therefore motivated to look for a new way of describing the process which has a higher predictive efficiency. That is, the desire to describe things simply makes us *look for* emergent behavior in self-organizing systems. (Imagine describing an excitable medium, not by saying where the spiral waves are centered and how their spirals curve, but by giving the complete field of molecular concentrations at each point.) Emergence without self-organization is definitely possible — for example, we've seen that thermodynamics emerges from statistical mechanics in a stationary (and so definitely non-self-organizing) system. I presume there can be self-organizing, non-emergent processes, though it *might* be that some constraint on possible ϵ-machines rules that out. Assuming, however, that self-organization does not imply emergence, then it is conceivable that there are processes which organize themselves into conditions so complex that no human being can grasp them. They would be so organized, in other words, that they would look very like noise. (Cf. Crutchfield and Feldman 2001a; Crutchfield and Feldman 2001b; Lem 1968/1983.) Emergence may be a pre-condition of *detectable* self-organization.

Engineering Self-Organization

The philosophy and theoretical explorations are definitely interesting and useful, but from the engineering perspective the focal concern is *not* which systems to call self-organizing and which not, but rather *how to create* systems that display those positive properties that people usually associate with self-organizing systems. This is currently quite an active area of research all over the world and no general and conclusive methodologies are available yet, but at least there is hope for major advancements within the next few decades, and some initial sketchy ideas from various researchers can already be presented here.

First of all, it would be helpful to have a more practical description of what an *artificial* self-organizing system is. Gershenson (2007) provides quite a suitable definition: "In engineering, a self-organizing system would be one in which elements are designed to dynamically and autonomously solve a problem or perform a function at the system level. In other words, the engineer will not build a system to perform a function explicitly, but elements will be engineered in such a way that their behaviour and interactions will lead to the system function.".

Why and When?

The next questions to ask would be "Why?" and "When?", that is, what exactly are the benefits of having some self-organizational processes in a system, and in which cases would it make sense to exploit them. These questions can be answered by taking a closer look at the typical good and bad characteristics of self-organization. It should be noted, though, that while the following features are fairly common in the stereotypical self-organizing systems (where interactions are strongly localized), they are not necessarily guaranteed to always occur, especially in the more advanced kinds of artificial self-organizing systems that sometimes do not adhere to the ideas of strong locality. The following discussion is partly based on Heylighen and Gershenson (2003), Prehofer and Bettstetter (2005), and Dressler (2007).

On the positive side, self-organizing systems tend to be intrinsically robust – able to withstand a variety of errors, perturbations, even partial destruction. This is partly because there are usually no single points of failure – local errors have, at least initially, only local consequences – and partly thanks to the processes like self-repair, homeostasis and dynamic resilience. The latter, in turn, means that self-organizing systems can be robust without being rigid, which allows for adaptivity. The locality of interactions also supports high scalability by having the addition of components somewhere in the system to not require much or any changing of the type and complexity of components' communication abilities. Similarly, in the other direction, we can expect *graceful* degradation.

On the negative side, strong locality may cause inconsistencies in the system (when different parts make different decisions), slow spreading of important information, and lower overall efficiency due to the lack of global optimization. And while scalability may be high in the spatial dimensions, it is rather problematic with regard to complexification and diversification of the system – adding new kinds of components and / or trying to combine multiple self-organizational mechanisms can lead to unforeseen effects. As of graceful degradation, it may be difficult to determine in advance within which limits it does remain graceful.

The increased autonomy that comes along with the ability to internally deal with the organizing processes can be positive – eliminating the need for direct control – but the obvious flip side is limited controllability of the system. Also, the behavior of the system, especially if we do not have a full understanding of it or lack the means to observe all influential interactions, can deliver us significant surprises, which, again, may be both positive and negative. A positive characteristic of self-organizing systems that is often considered important is the potential to get complex behavior from simple components, thus making the system, hopefully, easier and cheaper to design and manufacture. While certainly true in some cases, it is not a universal feature and there can be situations where the components would actually need to be *more* complex than in a traditionally engineered system in order to cope with everything by themselves, as there are no control elements to get the information and commands from. Moreover, no good and reliable engineering approaches are available yet for building self-organizational capabilities into artificial systems, let alone for combining many different such capabilities within one entity. And, additionally, the created artifacts may be difficult to test, verify, validate, and thus, to trust.

So, to figure out whether to use self-organization in some specific system, it is necessary to determine the negative and positive sides of it in that particular case and weigh them against each other and against other alternative approaches. Doing that analysis is, of course, not necessarily very simple.

How to Make Self-Organizing Systems?

Although no well-defined methodologies exist for building self-organizing systems, a lot of ideas and advice have already been put forward. A large part of them, though, are mainly concerned with the aforementioned stereotypical self-organizing systems where interactions are strongly localized.

First of all, the elements making up the self-organizing system need, obviously, to be able to interact with each other (Gershenson, 2007). But this does not always need to be full language based communication, nor even direct communication – for example stigmergy (indirect communication via making changes to the environment) is quite a common interaction method in some types of self-organizing systems. A prevalent suggestion, inspired by natural systems, is to keep the interactions local and, similarly, to make the behavioral rules of the components depend only on local information (Prehofer and Bettstetter, 2005), although in principle this may seriously limit the repertoire of the system and thus should not be taken as an undisputable commandment, especially given the elements of artificial systems have various capabilities not existing in their natural role models. Designing the interactions to be anonymous and uncoupled (i.e., when interaction partners need neither to know each other in advance, nor to be active at the same time; a typical example would be the aforementioned *stigmergy*) is also quite helpful in achieving the robustness and scalability associated with self-organization (Mamei et al., 2006). A similar strategy is to use dynamic discovery of communication partners and services, e.g., via broadcast (Zambonelli et al., 2004). Another possibility to limit dependencies is to replace, as much as possible, the use of long-lived information about others' / system's state with the use of real-time information, or at least to refresh it often (Prehofer and Bettstetter, 2005).

An important difference from most of the traditional engineering is the deliberate use of positive feedback where fluctuations get amplified and certain configurations or events are self-reinforcing (Camazine et al., 2001). For example, to achieve congregation it could be arranged that the individuals have a tendency to join a cluster if they see it, with the probability proportional to the size of the cluster. Then a mere accidental / random piling up of a few individuals somewhere would be enough to trigger an accelerating process of congregation. Of course, negative feedback still has an important role, too – when the positive feedback has done (enough of) its job, the negative one should balance it out. It can happen naturally, e.g., due to the exhaustion of some resource, or by some built-in trigger thresholds that cause the balancing mechanisms to kick in or intensify. Additionally, to achieve dynamic resilience, the positive feedback should not be turned off after its initial job is done, but left in this "capped" (balanced by negative feeback) condition, so that when some disturbance occurs it would immediately be able to do its restorative job again.

If the system is composed of identical elements in an (initially) homogeneous environment, then the inclusion of some randomness ("coin-flipping") can be helpful to break local symmetry and achieve more complex patterns and behaviors (Nagpal, 2006), or, more generally, randomness, noise and fluctuations can be exploited for allowing metastable systems escape their current basin of attraction and enter other, potentially better ones (Lucas, 2008). Further suggestions (all from Prehofer and Bettstetter, 2005) include the intentional toleration of conflicts as long as they are localized, restricted in time, and / or easily detected and resolved; not aiming for perfect and explicit coordination and, instead, trying to coordinate the components implicitly, based on the information that can be inferred from the local situation; and, where appropriate, making rules and protocols adaptive, so that the components would also cope with those changes that are not directly managed by the self-organizational processes.

Another feature that can be useful in spatially distributed systems is the ability of the components to find their location relative to others (Zambonelli et al., 2004) and then, possibly, behave differently depending on their position. In amorphous computing (Abelson et al., 2007) this is achieved by the self-generation of a local coordinate system, which can be done by various methods that are usually based on random symmetry breaking and gradient-producing mechanisms. The research on amorphous computing is actually doing quite an in-depth practically oriented work on the engineering of self-organizing systems and has generated a lot of other useful ideas as well. Some further approaches suggested by Abelson et al. (2007) include limiting certain behaviors to specific regions (that are created using coordinates, clustering, or some contextual information); getting inspiration from such physical and social concepts like fields, streams, gradients, diffusion, gossip; avoiding the reliance on global time by using relative times that each component can measure by itself, or no timing at all (event-driven systems); using trigger rules that locally detect the completion of a behavioral phase if wanting to compose a sequence of systemic behaviors (that is otherwise rather difficult to do with no global coordination); and, finally, trying to move the design and programming of self-organizing systems to incrementally higher levels of abstraction, similar to common programming languages (but probably using domain-specific abstractions). The latter, of course, requires proper understanding of how the lower levels work and how to automatically generate low-level code / components from those abstractions.

How to Get Them do Something Useful for Us?

Getting an artificial system to exhibit self-organization is surely an achievement by itself, but from the engineering viewpoint it is additionally necessary to ensure that this system will have some practical value. Again, hardly any methodology existing today could be considered well-defined and generally satisfying, but a lot of useful ideas and advice can be found nevertheless.

The most common approach is to copy (some aspects of) naturally occurring self-organizing systems by finding a suitable "role model", mapping its constituent parts to the new application domain, and adjusting the parameters of the resulting system until satisfactory behavior is achieved (Sudeikat and Renz, 2006). But the natural self-organizing systems are most probably only a tiny sampling of the possibilities that could be realized with naturally occurring components, and, additionally, our technology has capabilities that do not normally occur in nature, expanding the space of possibilities even further. So copying is just a small subset of what we could and should do, not the ultimate solution.

One alternative approach that does not require full understanding of how to purposefully construct useful self-organizing systems would be to use evolutionary design (Prokopenko, 2007) where variations of some initially unsuitable (and possibly simple) self-organizing system are generated repeatedly and better ones automatically selected to be the seeds for the next rounds of variation generation.

A somewhat similar idea, but applied *in* the design of a particular self-organizing system instead of *to* the design process on a meta-level, is to specify the wanted results (system's state or behavior) as the satisfaction of a set of constraints, and to build the rules to continually drive towards that satisfaction, as opposed to thinking only in a step-by-step-instructions mindset (Abelson et al., 2007). Or, alternatively, we could approach the problem from the other direction and *impose* the constraints within a functioning system so that undesired possibilities are blocked out. Other suggestions include trying to find the points of influence in the self-organizing system, where little interventions have large effects, and to use them for guiding the system towards desired behavior (Mikhailov and Showalter, 2008), and, in the development process, making extensive use of simulations, prototyping, testing, and revising, and also of applying mathematical analysis where possible. The latter can at times be forgotten because the existing mathematical apparatus tends to be insufficient for the complexities of larger practically interesting self-organizing systems, but it can nevertheless be a useful tool in our mental toolbox, to be used for some subtasks and for some levels of abstraction in the design process. Educative examples of mathematical analysis of naturally occurring self-organizing systems can be found in Camazine et al. (2001).

As the complexity of the desired functioning of the system grows, our ability to achieve it by constructing a self-organizing system out of nearly identical elements (that apparently has been the prevalent approach so far) drops sharply. To overcome this limitation, it has been proposed to move upward in design complexity and to allow for (fundamentally) different components within one system. For example, Zambonelli (2006) envisions a decentralized "ecosystem" that contains multiple coexisting "specimens" of complex self-organizing systems. To enforce at least some level of control over the ecosystem, he suggests to populate it with additional manager components that may (or may not, as necessary) look like native components to all others but have additional capabilities and try to guide the system by purposefully changing their behavior. Note that in general the managers do not always need to be separate system-level components – sometimes it might be more efficient to just add some managerial functionality into the already existing components, either as separate submodules or even in a fully intertwined way (depending on the level of autonomy of that managerial functionality, the latter may already be the realization of one of our main goals – an artificial self-organizing system where all the desired (complex) behavior arises directly from native components only). Similarly, Prokopenko (2007) proposes the use of hybrid systems where self-organization is limited to certain components or functionalities or levels of the system and the rest is engineered in the traditional way.

The Way Forward

In principle there are several ways to move towards the mastery of engineering artifical self-organizing systems. One would be to start from scratch and figure out and design the mechanisms with as much inventiveness and creativity as possible. This may yield very interesting and novel results, but is rather difficult. Another option would be to thoroughly study the already existing self-organizing systems and then, based on the gathered knowledge, start producing new insights and ideas. This is, in large part, how the engineering of self-organization has proceeded so far. And the best way would probably be to combine and use both (all) these approaches. But to do so more efficiently, and especially to speed up the second strategy, it would be most helpful to start composing a library (repository) where descriptions and analyses of naturally occurring self-organization examples, mechanisms and (very importantly) the spaces of possibilities of those mechanisms are collected. Along the way it should be expanded with new knowledge from artificial systems, preferably from systematic explorations of the possibilities, not only arbitrary examples of what has been done. The goal should be to acquire knowledge of what can *in principle* emerge from given *kinds* of systems, to become aware of their spaces of possibilities. Surely, it is impossible to do this exhaustively, to cover all potential mechanisms and emergent behaviors, but even a semi-systematic knowledge base of more thorough analyses of simpler systems together with some broader and partial descriptions of the possibilities of complex ones would be a huge step forward – a non-exhaustive library is still much better than no library at all. But, to get started, even random sampling would be good enough, and that indeed seems to be the current state of research on self-organizing systems (except that the results of this random sampling tend to be distributed over diverse sources and do not form an easily accessible collection).

Ideally, the field of engineering of self-organization will reach a stage where the designer will only need to specify the system in some high-level language and the technical details of how to create individual components that would collectively fulfill the given functions will then be figured out automatically. Building such global-to-local compilers is surely not trivial, but neither is it impossible, as already demonstrated in amorphous computing (see, e.g., the Proto language by MIT: http://groups.csail.mit.edu/stpg/proto.html).

References

H. Abelson, J. Beal, G. J. Sussman (2007), "Amorphous Computing", MIT Computer Science and Artificial Intelligence Laboratory Technical Report.

W. R. Ashby (1962), “Principles of the self-organizing system,” in Principles of Self-Organization: Transactions of the University of Illinois Symposium, H. Von Foerster and G. W. Zopf, Jr. (eds.), Pergamon Press, pp. 255-278.

S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau (2001), Self-Organization in Biological Systems, Princeton University Press.

F. Dressler (2007), "Self-Organization in Sensor and Actor Networks" (slides), http://www.tm.uka.de/itm/uploads/folien/177/selbstorganisation-20071113.pdf.

C. Gershenson (2007), "Design and Control of Self-Organizing Systems", PhD thesis, Vrije Universiteit Brussel.

F. Heylighen (2003), "The science of self-organization and adaptivity", in The Encyclopedia of Life Support Systems, L. D. Kiel, (ed.), EOLSS Publishers, Oxford.

F. Heylighen, C. Gershenson (2003), "The Meaning of Self-organization in Computing", IEEE Intelligent Systems, 18(4), pp. 72-75.

E. Karsenti (2008), "Self-organization in cell biology: a brief history", Nature Reviews, Molecular Cell Biology, 9(3), pp. 255-262.

C. Lucas (2008), "Self-Organizing Systems (SOS) FAQ", http://www.calresco.org/sos/sosfaq.htm.

M. Mamei, R. Menezes, R. Tolksdorf, F. Zambonelli (2006), "Case studies for self-organization in computer science", Journal of Systems Architecture, 52, pp. 443-460.

P. B. Medawar (1982), "Herbert Spencer and General Evolution", in Pluto’s Republic, Oxford University Press, pp. 209-227.

A. S. Mikhailov, K. Showalter (2008), "Introduction to Focus Issue: Design and Control of Self-Organization in Distributed Active Systems", Chaos, 18(2).

R. Nagpal (2006), "Self-Organizing Shape and Pattern: From Cells to Robots", IEEE Intelligent Systems, 21(2), pp. 50-53.

C. Prehofer, C. Bettstetter (2005), "Self-Organization in Communication Networks: Principles and Design Paradigms", IEEE Communications Magazine, 43(7), pp. 78-85.

M. Prokopenko (2007), "Design vs Self-organization", in M. Prokopenko (ed.), Advances in Applied Self-Organizing Systems, Springer, pp. 3-17.

C. R. Shalizi (2001), "Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata", PhD thesis, University of Wisconsin at Madison.

C. R. Shalizi (2009), "Self-Organization", http://cscs.umich.edu/~crshalizi/notebooks/self-organization.html.

C. R. Shalizi, K. L. Shalizi, R. Haslinger (2004), "Quantifying Self-Organization with Optimal Predictors", Physical Review Letters, 93(11).

J. Sudeikat, W. Renz (2006), "On the Redesign of Self-Organizing Multi-Agent Systems", International Transactions on Systems Science and Applications, 2(1), pp. 81-89.

Wikipedia (2010), "Self-organization", http://en.wikipedia.org/w/index.php?title=Self-organization&oldid=339878193.

F. Zambonelli (2006), "Self-Management and the Many Facets of 'Nonself'", IEEE Intelligent Systems, 21(2), pp. 53-55.

F. Zambonelli, M.-P. Gleizes, M. Mamei, R. Tolksdorf (2004), "Spray computers: frontiers of self-organization for pervasive computing", 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04).