

# Self-Organizing Networked Systems for Technical Applications: A Discussion on Open Issues

Wilfried Elmenreich<sup>1</sup> and Hermann de Meer<sup>2</sup>

<sup>1</sup> Lakeside Labs, Mobile Systems Group  
Institute of Networked and Embedded Systems  
University of Klagenfurt, Austria  
wilfried.elmenreich@uni-klu.ac.at

<sup>2</sup> Faculty of Informatics and Mathematics  
Chair of Computer Networks and Communications  
University of Passau, Germany  
demeer@uni-passau.de

**Abstract.** The concept of self-organization has been examined oftentimes for several domains such as physics, chemistry, mathematics, etc. However, the current technical development opens a new field of self-organizing applications by creating systems of networked and massively distributed hardware with self-organized control. Having this view in mind, this paper reviews the questions: *What is a self-organizing system?*, *What is it not?*, *Should there be a separate field of science for self-organizing systems?*, and *What are possible approaches to engineer a self-organizing control system?*. The presented ideas have been elaborated at the *Lakeside Research Days'08* (University of Klagenfurt, Austria), a workshop that featured guided discussions between invited experts working in the field of self-organizing systems.

## 1 Introduction

The idea of Self-Organizing Systems (SOSs), although long known from domains such as physics, chemistry, and biology, has recently gained interest to be applied to technical applications. The reason for this is a paradigm shift from monolithic systems or systems with a small number of components to large networked systems. This paradigm shift is driven by the technological advancement and the emergence of pervasive systems integrating information processing into everyday objects and activities. For example, a fieldbus network with accurate but expensive sensors interconnected by a dependable wired communication system might be replaced by a system of hundreds of small, but inexpensive sensors using a wireless ad-hoc network to interconnect. Such a cyber-physical system [1] can use the view of multiple sensors to come to a massively distributed view of a technical process, where the fusion of several sensor measurements potentially leads to a

more extensive, more accurate, and more robust observation. However, such an approach requires a control paradigm that copes with the complexity of such a solution. A promising approach to attack this problem is the principle of self-organization, where the control is as well decentralized as the controlled system. Through the definition of the behavior in local interactions, it is expected that the overall system shows emergent behavior such as complex order or properties like robustness, adaptability and scalability.

Designing and controlling an SOS can be very demanding. There is no general methodology yet explaining how to design an SOS and in many cases it is very difficult to provide a concise validation of the system. In order to identify the current problems and to search for possible directions for a solution, the Lakeside Labs at the University of Klagenfurt, a research center focusing on networked self-organizing systems, arranged a one-week event called *Lakeside Research Days'08* by inviting international researchers working in the area of SOSs to discuss the topic of SOSs and its open problems. It is the purpose of this paper to summarize the main results of the workshop in order to give researchers an idea of the open problems and the potential for future research.

The rest of the paper is organized as follows: Section 2 briefly reviews the results of a discussion on the definition of SOSs followed by identifying several misconceptions on the understanding of SOSs. Section 3 approaches the question if SOSs should become a separate field of science having its own experts, terminology, and nomenclature. The question how an SOS can be designed is addressed in Section 4 by sketching three different types of design approaches. Section 5 concludes the paper.

## 2 Definition and Misconceptions of Self-Organizing Systems

In order to communicate problems, methods and results, it is necessary to have a common understanding of the terms used in scientific communication. The term “self-organization” is used by many researchers, but it has no generally accepted meaning. Instead, there exist a number of definitions from different domains such as from cybernetics [2, 3], mathematics [4], information theory [5], etc. Gershenson and Heylighen [6] argument to evade the debate about an exact definition of SOSs and to regard self-organization merely as a *way* of observing systems. Depending on the system type and its applications, understanding a system as being self-organized can be more or less helpful.

Therefore, instead of trying to find a single concise definition of SOSs we worked out a brief sentence explaining the main idea of SOSs that assists in communicating the main idea. Followed by this, we elaborated on the common misconceptions in the definition of SOS. Among several proposals, the following sentence, which was an outcome of the discussions at the Research Days [7] showed an interesting attempt to sketch the concept of SOSs in a nutshell:

*A self-organizing system (SOS) consists of a set of entities that obtains an emerging global system behavior via local interactions without centralized control.*

In the following, we address the question of common misconceptions in the understanding of SOSs.

### **2.1 Misconception #1: Self-organizing systems establish a class of systems**

If a system is considered to be self-organizing or not depends mainly on the way how the system is observed, especially where the borderline between the observed system and its environment is drawn. Gershenson proposes the following perspective to overcome this problem: Instead of thinking of SOSs as an absolute class of systems, self-organization should be understood as a way of observing systems [6]. Depending on the type of problem and the desired solution, the way of observing a system as an SOS can be beneficial or not.

### **2.2 Misconception #2: Self-organizing systems are chaotic systems**

There is a relation between chaos theory and self-organization in that a SOS may show chaotic behavior, that is having critical turning points (also known as bifurcations) in the system behavior [8]. However, an SOS does not necessarily have to show such behavior. Instead, some SOSs also might approach their target state without a sensitive dependence on initial conditions. Accordingly, a system with chaotic behavior may be built without employing the typical building blocks of SOSs such as distributed entities and local interactions.

### **2.3 Misconception #3: The emerging structure is a primary property of self-organizing systems**

SOSs provide a powerful mechanism to create structure and patterns. This phenomenon can be observed in many physical and biological systems, such as the skin pigmentation of fish, the polygonal pattern of nest territories of fish such as *Tilapia*, or the cathedral-like buildings of termites [9].

However, the emerging pattern should not be seen as a primary property of an SOS. There are SOSs, like homeostatic operational control in living beings, where such a structure is not present or is hidden from the observer. Thus, the emerging structure can be rather seen as a secondary property that indicate self-organization in many cases.

### **2.4 Misconception #4: Self-organizing systems are always based on evolutionary processes**

Evolutionary processes, as best known from biological examples, are an iterative mechanism of change in the inherited traits of a population of organisms from one generation to the next [10]. Evolutionary processes are driven by mutation, selection and recombination.

Many biological examples of self-organizing systems have emerged from an evolutionary process, which made the term self-organization connected to evolution. Thus, the connection of SOSs to evolutionary processes is not an obligatory one, since many non-biological examples of SOSs have developed without an evolutionary process, thus showing the possibility to design self-organizing without an evolutionary process. However, an interesting research task for future technical systems arises in constructing SOSs, which implement an evolution of their local rules in order to adjust to new situations.

### **2.5 Misconception #5: A self-organizing system never needs maintenance**

Many SOSs show adaptive behavior, which means that they can operate well within a wide range of input parameters. However, that does not imply that a technical SOS will have a low maintenance effort. Typically, a complex technical system that must operate over a considerable life time will require maintenance in order to provide its service during system lifetime. It is an open question if maintenance of a technical system with self-organizing properties will be easier or more complicated to maintain than a traditionally designed technical application. On the one hand, properties like robustness might make it easier to replace parts of the system without disturbing the overall operation, on the other hand, diagnosis and maintenance of an SOS might turn out to be more complex than in systems built following a more straightforward approach.

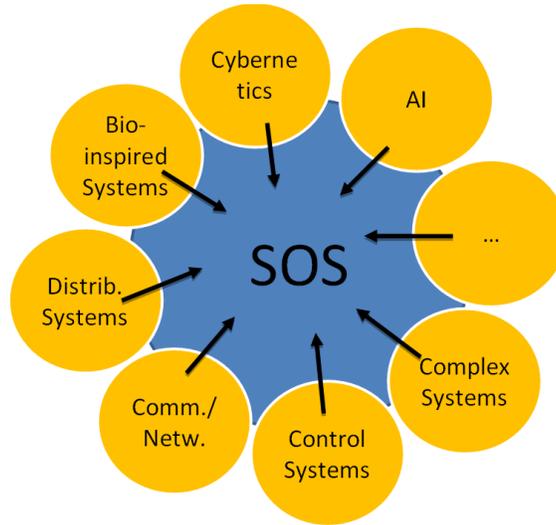
## **3 Self-Organizing Systems Forming an Own Field of Science?**

A science field is characterized as a category of specialized expertise within science, often also having its own terminology and nomenclature. In the example of SOSs we cannot speak of a separate field of science today.

As sketched in Figure 1, SOSs are found in multiple disciplines, being thus a highly interdisciplinary field. This does not necessarily hinder the formation of an own field of science, as it is shown by the also highly interdisciplinary research on Artificial Intelligence, which is regarded as a separate field of science. A key question is, if the research on SOSs is likely to converge in a separate field of science with its own experts, terminology, and nomenclature. The other option is that research on SOSs will rather be covered by researches coming from one of the related fields.

The answer of the question is of great interest, since it will influence future decisions on installing academic curricula for the “field” of SOSs. We do not claim that we can give an absolute answer to this question, but we can formulate two subquestions to this issue:

*Is there a common vision within the field?*



**Fig. 1.** Interdisciplinary role of self-organizing systems

For example, Artificial Intelligence (AI), also consisting of many different methods and viewpoints, came at least with a common vision as it was presented in the proposal for the Dartmouth Summer Research Project by McCarthy, Minsky, Rochester and Shannon in 1955 [11].

*Are there fundamental research questions that would require investigation from dedicated SOSs researchers?*

If not, trying to force the creation of a new field with its own terminology and nomenclature would be an effort that can even worsen the communication between other domains since the problem is not discussed in the domain-specific language.

## 4 Design Approaches for Self-Organizing Systems

In [12], Prehofer and Bettstetter identify the task of finding *local behavior rules that achieve global properties* as a major paradigm to be approached. In the following, we will elaborate on design approaches for solving this problem.

### 4.1 Bio-Inspired Design

In nature, there are several examples of self-organizing behavior, for example, ants cooperatively finding shortest routes to food sources, termites building complex constructions without using a blueprint, fish schools organizing themselves without a leader, and swarms of fireflies in south-east Asia synchronously emitting light flashes.

There are two main paradigms of bio-inspired design: the direct and the indirect approach. In the direct (or top-down) approach, a technical problem is tackled by looking for natural examples solving an equivalent problem. The biological solution and its principles are then analyzed and re-built in a technical application. Examples of the direct approach are the design of aeroplane wings by observing the gliding flight of birds as it was done by Otto Lilienthal in the 19th century, or, after a closer analysis of the up-bent feathers at the wingtips of several birds, the refinement of aeroplane wings by turbulence-reducing and thus fuel-saving winglets [13].

In contrast, the indirect (or bottom-up) approach of bio-inspired design involves first the derivation of principles by analyzing natural systems. This step is done in a basic research effort that is not yet targeted at a specific application. The principle is then abstracted from its biological context and used in particular technical applications where they could be suitable. Examples of the indirect approach are the concept of artificial neural networks or the concept of ant foraging behavior being applied to mesh network packet routing.

An important aspect of bio-inspired design are the notable differences between biological and technical solutions: In many biological systems, especially for lower animals, there is no real counterpart to what we call software in technical systems. For example, protozoa have no mechanism to learn and circulate behavior during lifetime. Instead, new behavior is stored via the genes of the next generation. With the same effort, also physical changes of the next generation are possible. Thus, in such an evolutionary approach, the physical body and the physical abilities typically grow as part of the solution, while in traditional engineering the analogy of the body, that is the hardware, has to be usually fixed early in the design process. For example, a hardware revision comes with considerable little effort in biological systems while in a technical project an unscheduled hardware change likely might cause the project to go over budget. On the other hand, when using the biological approach only with the software part being able to change, the result might be less effective than the biological example. For example, a bio-inspired algorithm optimizing only the software for a mobile robot would not play with optimizing the number and placement of the robot's physical sensors while the biological counterparts co-optimize these aspects as well.

On the other hand, evolution is at a disadvantage when it comes to the creation of radically new designs. All living beings follow more or less a general blueprint, in case of a particular class such as mammals, the design is even more restricted. Thus, for example, nature was never able to design a wheeled vehicle-like animal or other rotating machines.

Thus, bio-inspired mechanisms might be inappropriate when (i) the biological solution is too difficult to rebuild with technical means or (ii) a technical solution can be more efficient by taking advantage of mechanisms that cannot be found in the biological paradigm. A mistake to be avoided is to stick to biological solutions just because of their seeming elegance.

These differences between nature's way to build things and engineering make especially a direct design approach very difficult to apply. For designing self-

organizing technical systems, the indirect approach seems more promising, since it allows the assignment of biological ideas in a wider context.

## 4.2 Trial and Error

This approach requires to have a testbed that allows extensive testing without high cost or possible endangering of persons. 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 in order to create trust in the derived solution.

The process of trial and error itself can be refined in several ways.

Gershenson [14] introduces the notion of *friction* to describe the fitness of an interaction between two entities. There exists also a friction of the overall system that should be minimized in order to maximize its performance. By identifying and analyzing points of friction, an engineer can change the local rules in order to mitigate the problem. However, this is not a straightforward approach – in several cases a higher friction for a particular entity might be beneficial for the overall system.

Furthermore, existing search algorithms can be applied to the search for an optimal or sufficiently good set of local rules. However, the search space is typically too large for an exhaustive search and its non-monotonic properties can get a non-exhaustive search algorithm to be stuck in local maxima. In this cases, heuristic search algorithms like genetic algorithms and simulated annealing can be a choice.

## 4.3 Learning from an Omniscient Solution

This approach requires to have an optimal or at least well-performing solution to the problem beforehand. However this solution might be created using features that cannot be realized in the final solution. An example could be the implementation of an omniscient algorithm for 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.

An example for such an approach is given by Auer, Wüchner and De Meer [15] by designing an agent performing well in the prisoner’s dilemma [16]. In the design process, an agent having perfect knowledge is created first, then its behavior is analyzed using Causal State Splitting Reconstruction [17], which is basically a method for time series analysis. The results are then used for designing the local rules for a non-omniscient agent. The resulting agent showed to be an improvement of the standard and already well-performing *tit-for-tat* strategy by implementing also forgiveness.

## 5 Summary and Conclusion

The connection of self-organization to so many disciplines in science is an advantage and a disadvantage at the same time. In terms of definition and terminology, the many definitions from different domains have blurred the overall idea which is definitely a disadvantage. On the other hand, the many disciplines keep the potential for many ideas and new approaches for creating self-organizing control systems. This possibility will be even more attractive, if the research on SOSs can converge towards a more standardized nomenclature, probably even forming a new field of science some day.

Several positive effects from the interdisciplinarity of self-organization became apparent when discussed possible ways to design the behavior of the particular entities. The local behavior is an integral part of an SOS, since the overall behavior of the system emerges from the local interactions of the entities. We have identified three basic approaches for finding a suitable set of local rules, namely bio-inspired design, trial and error, and learning from an omniscient solution. Bio-inspired design can give promising results, however, due to the differences in natural evolution and traditional engineering, it has to be applied carefully. The approach of learning from an omniscient solution might be also interesting for non-game-theoretic settings. However, the approach relies on the possibility that the omniscient solution can be build in a simulation and that the behavior can be mimicked by an agent with local information in a useful way. Trial and error is definitely the most general approach among the three. However, despite of improvements in identifying friction and search algorithms this approach can be too inefficient so that it may not succeed for a large search space. The right design approach (which may be also a combined approach) for a particular project will be defined by the particular constraints and requirements.

We hope that the discussions and suggestions in this paper will be helpful to future research in the area of SOSs. In the future, we plan on elaborating the design process for building SOSs by combining existing engineering approaches with methods especially tailored to SOSs.

## Acknowledgments

This work was supported by the Lakeside Research Days'08 under contract No. KWF-20214/16476/23861, taking place from June 30, 2008 to July 4, 2008 at the University of Klagenfurt, Austria. We would like to thank all participants of Lakeside Research Days'08 for the fruitful discussions on self-organization. Furthermore, we would like to thank Patrick Wüchner for constructive comments on the paper.

## References

1. E. A. Lee. Cyber physical systems: Design challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, January 2008.

2. H. von Foerster. On self-organizing systems and their environments. In M. C. Yovitts and S. Cameron, editors, *Self-Organizing Systems*, pages 31–50. Pergamon Press, 1960.
3. 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.
4. G. G. Lendaris. On the definition of self-organizing systems. *Proceedings of the IEEE*, 52(3):324–325, March 1964.
5. H. Haken. *Information and Self-Organization – A Macroscopic Approach*. Springer, 1988.
6. C. Gershenson and F. Heylighen. When can we call a system self-organizing? In *7th European Conference on Advances in Artificial Life, ECAL'03*, volume LNAI 2801, pages 606–614. Springer Verlag, 2003.
7. W. Elmenreich. Lakeside Research Days'08. Technical report, Lakeside Labs, Klagenfurt, Austria, June-July 2008.
8. S. L. Bloom. Chaos, complexity, self-organization and us. *Psychotherapy Review*, 2(8), August 2000.
9. S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, 2001.
10. Wikipedia, the free Encyclopedia. Evolution. Wikimedia Foundation, August 6 2008.
11. J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence. Technical report, Dartmouth College, 1955.
12. C. Prehofer and C. Bettstetter. Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine*, pages 78–85, July 2005.
13. R. Faye, R. Laprete, and M. Winter. Blended winglets. *Aero, Boeing*, (17), January 2002.
14. C. Gershenson. *Design and Control of Self-organizing Systems*. PhD thesis, Vrije Universiteit Brussel, 2007.
15. C. Auer, P. Wüchner, and H. de Meer. A method to derive local interaction strategies for improving cooperation in self-organizing systems. In *Proceedings of the Third International Workshop on Self-Organizing Systems*, Vienna, Austria, December 2008.
16. A. Tucker. *A two-person dilemma*. Stanford University Press, 1950.
17. C. R. Shalizi and K. L. Shalizi. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In M. Chickering and J. Halpern, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*, pages 504–511, 2004.