How to Design Self-Organizing Systems?

Extended Abstract

Wilfried Elmenreich
Lakeside Labs, Mobile Systems Group
Institute of Networked and Embedded Systems
University of Klagenfurt
Klagenfurt, Austria
wilfried.elmenreich@uni-klu.ac.at

Gerhard Friedrich
Intelligent Systems and Business Informatics Group
Institute of Applied Informatics
University of Klagenfurt
Klagenfurt, Austria
gerhard.friedrich@uni-klu.ac.at

ABSTRACT

The behavior of a self-organizing system (SOS) is typically defined by the local interaction rules of the components. While this emergent behavior typically is very flexible, i.e., working at different scales being robust against disturbances and failures, there exists no straight-forward way for the design of these rules so that the overall system shows the desired properties. The try and error methods, even when being improved using notions such as the "friction" between two components often suffer from counter-intuitive interrelationships between local rules and emergent behavior. Imitation approaches, such as the bio-inspired methods or the programming of the local behavior by analyzing an example using perfect knowledge are limited to the cases where an appropriate example model is available. Therefore, we investigate on novel generic approaches for designing self-organizing systems. E. g., promising methods could be genetic algorithms or particle swarm optimization methods. A possible approach to model the local interaction in a way that can be evolved is given by neural networks. In the long term, we aim at a generic optimization tool for designing and exploring rulesets for SOS designs.

Keywords

Evolution-inspired systems and interactions, complex adaptive systems, self-organizing systems

1. INTRODUCTION

In a discussion among experts on SOS at the Lakeside Research Days 2008, the design of local rules to achieve a desired global behavior was ranked top among the most important problems for the design of SOS systems. In [5], Prehofer and Bettstetter also identify the task of finding *local behavior rules that achieve global properties* as a major paradigm to be approached.

Finding a working set of local behavior rules is often a very complicated task, due to the fact that the emergent service is difficult to predict from the local rules. Changes to the local rules even might lead to counter-intuitive results as depicted by M. Resnick in [6] by the example of a simple simulation of slime mold behavior: A majority of researchers, among them many experts on self-organized behavior, being asked for the effect of a simple change in the local rules were predicting the wrong result.

Therefore, we identified a high relevance for elaborating design methods for self-organizing systems. Currently, there exist three approaches for finding a suitable set of local rules, namely bio-inspired design, learning from an omniscient solution, and trial and error. Bio-inspired design derives its rules from a biological example, as it is been the case with the Firefly synchronization example [3]. Bio-inspired design can give promising results, however, it requires the existence and discovery of a biological example solving the problem in question. Furthermore, due to the differences in natural evolution and traditional engineering, it has to be applied carefully. An example for learning from a given omniscient solution is provided by Auer, Wüchner and De Meer [1] by designing an agent performing well in the prisoner's dilemma [9]. In the design process, an agent having perfect knowledge is created first, then its behavior is analyzed using Causal State Splitting Reconstruction [7], which is basically a method for time series analysis. The results are then used to design the local rules for a non-omniscient agent. Trial and error is definitely the most general approach among the three. However, the large parameter space of possible settings and the often counter-intuitive effects require an appropriate search strategy.

A promising approach for designing the local rules without explicit knowledge of the effect of a particular ruleset on the emerging service is the combination of a genetic algorithm and a simulation of the target system. The genetic algorithm is used to evolve the local behavior rules towards the maximum of some fitness value, which is derived from simulation runs where the ruleset under consideration controls the local entities. An example for this approach could be the optimization of a group of mobile robots to cooperate in searching a floor as a global task. The behavior parameters of the single robots will then be evolved using selection, mutation, and inheritance for the purpose of achieving the best fitness in the simulation. Note that the best solutions might include strategies, where particular robots have to sacrifice their local performance for a better overall result. Such counter-intuitive anomalies make it difficult for finding a good overall solution by manual try-and-error approaches. A constraint for such an evolutionary approach is that the representation of the local behavior rules must be evolvable, i. e., it must be possible to apply mutations and inheritance in to the program with a fair chance of keeping the overall program behavior. In some cases, genetic algorithms have been applied to evolve artificial artificial neural networks (ANN) [10, 2]. Sipper [8] shows the versatility of this approach by applying it to different game playing problems. There exists almost no related work aiming specifically on the evolution of cooperative systems or self-organizing systems based on ANN controllers. A notable exception is the work of Nelson [4], describing the evolution of multiple robot controllers towards a team that plays "Capture the flag" against an opponent team of robots.

Thus, while genetic algorithms and neural networks are known and in use for some time, there exists a relevant and interesting research challenge in applying them for designing self-organizing systems. Furthermore, this project will aim at providing the technology, in form of a reference architecture and a tool implementation, for designing self-organizing systems generically.

2. RESEARCH CHALLENGES

Representation of local behavior rules: Self-organizing behavior is induced by the local interaction of the micro-components of the system. While there exist several efforts in identifying and classifying types of self-organizing systems, the representation of the local rules has not received much attention in the literature. Thus, we see a need to investigate possible representations for local behavior rules of self-organizing systems and classify them according to the implications of using a particular representation. Our hypothesis is that there is not a single rule language that fits all applications but that the selection of the appropriate representation is an important and delicate task in the design process of a self-organizing system. The type of rule representation is expected to have a major influence on properties like the comprehensibility, evolvability, and verifiability of a system.

Methodology for designing self-organizing systems: Current design methods and tools are either limited to particular constraints or not feasible because of the complexity and effort of the approach. A possible generic tool could be based on an evolutionary approach to find a solution for a problem which cannot be directly solved, which is is typically the case for self-organizing systems.

Validation and verification: For the real world applicability of self-organized multi agent systems the central question is how to guarantee certain properties of the overall system. However, especially for self-organizing systems it becomes often difficult to predict or guarantee a particular global outcome based on the local rules. In order to have trust in such a system, appropriate validation and verification techniques must be applied. For example, verification techniques based on theorem proving (e.g. model-checking) could be used to guarantee specific system properties. An other approach is the combination of theoretical analysis with data from practical evaluations in the field.

3. OUTLOOK

We see a strong need to investigate on the research challenges stated above. A further meta-challenge is given by the fact that knowledge from different areas, such as complexity theory, genetic algorithms, neuroinformatics, multi-agent systems, etc. is needed in order to work successfully on these challenges. Therefore, a good cross-linking to several scientific communities is a key to success.

Another important aspect is that research in this area may not stop at a theoretical level. Self-organizing systems show a high potential to solve engineering problems concerning complexity, robustness, adaptivity and scalability. However, there is a need to transform this potential into the technology which can be used in future intelligent devices and products.

4. ACKNOWLEDGMENTS

This work was supported by Lakeside Labs GmbH in Klagenfurt am Wörthersee, Austria. Lakeside Labs is a research center investigating self-organizing networked systems: `http://www.lakeside-labs.com`

5. REFERENCES

[1] 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, Dec. 2008.

[2] D. Floreano and F. Mondada. Evolving of homing navigation in a real robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(3):396–407, June 1996.

[3] R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, Dec. 1990.

[4] A. L. Nelson, E. Grant, and T. C. Henderson. Evolution of neural controllers for competitive game, 2004.

[5] C. Prehofer and C. Bettstetter. Self-organization in communication networks: Principles and design paradigms. *IEEE Communications Magazine*, pages 78–85, July 2005.

[6] M. Resnick. *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds (Complex Adaptive Systems)*. The MIT Press, 1997.

[7] 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.

[8] M. Sipper, Y. Azaria, A. Hauptman, and Y. Shichel. Designing an evolutionary strategizing machine for game playing and beyond. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(4):583–593, 2007.

[9] A. Tucker. *A two-person dilemma*. Stanford University Press, 1950.

[10] X. Yao and Y. Liu. Evolving artificial neural networks through evolutionary programming. In *Proc. of the Fifth Annual Conference on Evolutionary Programming*, pages 257–266. MIT Press, 1996.