Discussing Whether the System in "Distributed Multi-Objective Scheduling of Power Consumption for Smart Buildings" Is in the Field of Organic Computing

Torben Keller

July 23, 2023

Discussing whether the system in ”Distributed multi-objective scheduling of power consumption for smart buildings” is in the field of Organic Computing

1st Torben Keller
Christian-Albrecht-Universität zu Kiel
Kiel, Germany
stu217932@mail.uni-kiel.de

***Abstract*—The energy transition towards renewable energy sources requires a distributed power grid with many small renewable low voltage power plants that adjust their energy output to the power consumption. Marvin Nebel-Wenner et al. proposed in 2019 a distributed system to reduce the overall energy prices in a distributed power grid. This paper investigates whether that system has the necessary robustness, adaptivity and manageability using concepts from Organic Computing and proposes suggestions for improvement.**

***Index Terms*—organic-computing, self-organizing-systems, power-consumption**

I. Introduction

The energy transition towards renewable energy sources requires a new distributed structure of the power grid. Instead of a few big high voltage power plants (i.e. coal or gas power plants) the future power grid will consist of many small renewable low voltage power plants (i.e. photovoltaic or wind turbines). Also the electrification of vehicles and the heat transition in the building sector towards heat pumps will increase the overall demand of electricity. This is a big challenge for the power grid, because renewable power plants might produce power when there is no demand, or produce no power when the demand is high [1].

To keep the power grid robust, adaptive and manageable the power plants and consumers have to coordinate the power supply, demand and storage usage. A self-organizing system approach is needed for the coordination where agents are power plants and consumers would enable this coordination in a robust, adaptive manageable way. Marvin Nebel-Wenner et al. simulated such a system in [1].

Organic Computing (OC) proposed concepts to reduce complexity and increase robustness of distributed systems such as self-adaption and self-organization [2]. These are mandatory properties for a system that schedules power consumption in a distributed power grid. This leads to the question whether the presented system is an instance in the domain of organic computing and fulfills the mandatory requirements.

The remainder of this paper is organized as follows: Section 2 briefly summarizes organic computing and self-organization. Section 3 describes the system used in [1]. Section 4 discusses whether can be classified as a system in the domain of organic computing. Finally the paper closes with a summary in Section 5.

Background

In 2017 Tomforde et al. updated the definition of the term “Organic Computing” and the desired self-attributes [2]. The term “organic” in the context of OC is meant as equipping computing systems with “life-like” properties such as autonomy of subsystems and robustness against disturbances.

An “Organic Computing System” (or organic system) is a technical system that adapts autonomously and dynamically to the current conditions of the perceived environment. Usually, an organic system consists of autonomous subsystems, each containing a part for the adaption aspects and a part to fulfill its utility. Organic systems do not always perform better in aspects such as higher speed or better quality of decisions than non-organic systems. Meaning the concepts of OC are no self-purpose. OC supports systems to become more robust and reduces complexity. The system’s degree of autonomy can range from total autonomy to no autonomy, with intermediate degrees of internal and external control. Neither total nor no autonomy are desirable, because either the system does not profit from autonomy, or the system becomes uncontrollable from outside.

The defined self-* properties of organic systems are:

1) **Self-configuration / Self-adaptation:** The organic system adjusts its parameters to fulfill higher-level user goals, enabling to exhibit desired behaviors.
2) **Self-organization:** The organic system dynamically adjusts its structure based on active user goals at runtime.
3) **Self-integration:** The organic system autonomously adapts its role, behavior and relations within a larger system for proper functionality.
4) **Self-management:** An collective term that contains self-configuration, self-organization and other self-* mechanisms

5) **Self-healing:** The organic system is capable to detect, diagnose and repair failures.
6) **Self-protecting:** The organic system defends itself as well as the overall system against attacks from outside and large-scale cascading failures that the self-healing mechanism is not capable to handle.
7) **Self-stabilizing:** The organic system strives to maintain stable behavior despite continuous adaptation or external influences.
8) **Self-improving / Self-optimization:** The organic system continuously analyses its decisions in organization, configuration and others seeking for better solutions.
9) **Self-explaining:** The organic system reasons its decisions on request and must ask for help to stay controllable by the human.

While these are the self-* properties defined in [2], the literature is very diverse regarding the properties forming a system in the domain of OC [2]. But in this paper we will use this definition.

In 2017, Sven Tomforde et al. introduced a measuring method to measure the degree of self-organization in systems by observing the systems communication [3].

In this paper, self-organization is defined as the continuous evolution of the systems structure caused by independent subsystems (agents) in order to fulfill the systems goal. If the subsystems adapt their behavior only by changing the systems goal, they are called autonomous.

The system model presupposes there is a potentially large amount of autonomous subsystems. Conceptually an autonomous subsystems internal structure consists of a productive system (PS) for the basic purpose and a control mechanism (CM) that knows the user defined goal of the system, controls the behavior of PM and decides about the relations of the subsystem. This would be autonomous, because actual decisions are taken by CM. The user can not step into the decision making process. In the system model the system composition is not restricted. We assume a set of agents as blackboxes where we may not have full access to each agent, the agents may belong to different authorities, or the agents can be controlled by different users. But we must be able to observe the actions taken and messages sent and received by the agents.

Technically relationships are functional connections. We assume establishing and changing relations requires communication, modeled as sending and receiving messages. The communication must go through a shared channel using standard protocols where agents are uniquely identifiable and messages visible.

To measure the degree of self-organization, the communication behavior of the system gets observed. The communication reflects the dynamics of self-organization. If self-organization happens due to disturbances, internal or external conditions, or modification of the utility function, the communication behavior of agents change over time. This means the degree of self-organization can be quantified by a divergence measure of two density functions $p(x)$ representing the earlier point in time and $q(x)$ representing the current observation cycle of the systems communication. The used divergence measure is derived from the Kullback Leibler (KL) divergence and results in following formula:

$$\begin{aligned} KL_2(p,q) =& \frac{1}{2}(-\int p(x)\ ln\ p(x)\ dx - \int p(x)\ ln\ q(x)\ dx \\ &+ \int q(x)\ ln\ q(x)\ dx - \int q(x)\ ln\ p(x)\ dx) \end{aligned}$$

$KL_2(p,q) = 0$ means no self-organization happens. $KL_2(p,q) > 0$ means self-organization happens and the larger $KL_2(p,q)$ is, the more self-organization happens [3].

Approach

The discussed system [3] models the power grid of the future with the ISAAC software, a unit aggregation and planning software based on the heuristic COHDA (Combinatorial Optimisation Heuristic for Distributed Agents). COHDA, presented by Hinrichs and Sonnenschein in [4], is using self-organizing mechanisms to optimize a common target. In [3] the common target is minimizing the electricity prize by using different power demand / supply schedules of smart buildings during the day. Each schedule also has a behavior adaption cost, prizing the adaption of the power consumption of the residents, which is added to the electricity prize.

Each smart building is simulated by two agents. One battery agent simulating the battery storage of a smart building and one building agent simulating the other electrical devices of the building. The authors assigned each building a feasible set of schedules, depending on the expected behavior of the residents, the generated energy of the photovoltaic plant and the size of the battery in the building. Each agent knows the feasible set of schedules for its associated building and selects one schedule that gets added to the solution candidate. The solution candidate is public to other agents while the feasible set of schedules is kept private.

The algorithm then consists of three steps:

1) **Perceive:** A message is the solution candidate of the sending agent. When an agent receives a message from one of its neighbors, it updates the information about the planned energy consumption of other agents and replaces the existing solution candidate, if the new candidate contains more elements (a new agent got added), or yields a better rating (the neighbor found a better solution candidate).
2) **Decide:** The agent then searches for the best of the feasible schedules of its unit. If the resulting system state yields a better rating a new solution candidate is created, which replaces the old one.
3) **Act:** If the solution candidate has been modified, the agent sends the new solution candidate to its neighbors.

Eventually all agents will store the same solution candidate which represents a local optimum of power consumption in this interval. This also terminates the algorithm.

To monitor and assuring termination within a desired time, the ISAAC software also adds an observer and a controller agent [1].

Discussion

In order to determine whether the system described in [3] is an instance of the domain of OC we will go through the defined self-* properties by Sven Tomforde, Bernhard Sick and Christian Müller-Schloer (2017) and check whether the system has the self-* property.

Self-configuration: The agents influence their environment through different mechanisms. The first by changing the demand causing the electricity price to change is negligible for one agent, because the electricity market is too big that one smart building alone could rise or fall prices. The other one is by propagating their solution candidate. Other agents will observe this information to adjust their own solution candidate and modify the environment themselves. In conclusion the system is self-configurable.

Self-organization: We can analyze the degree of self-organization using the communication behavior of the system as discussed in [3]. The more the communication changes over time, the more self-organization happens. Because we have no communication data of the system, we can not quantify the self-organization using the proposed divergence measure and have to discuss the change of density over time conceptually.

The agents notify their neighbors when the solution candidate changed, otherwise they will not communicate. Unfortunately the paper does not define what a neighbor is. Due to the small amount of participating agents, a broadcast is possible. In the context of buildings also the physical neighbors are possible. Because of the not explained self-configuration mechanism, the set of all known agents for each agent is a potentially smaller set than the set of all agents. In the following we assume no agents are added over time and the agents use a broadcast mechanism to notify other agents about a new solution candidate. The other possible solutions would potentially create partitions in the network resulting in a not properly propagated solution candidate and potentially not observed messages by the observing agent.

Using this assumption each agent either sends a broadcast, or nothing when perceiving a new message, depending on the observed environment. Assuming there is an optimal solution, eventually the system will find it resulting in a static system. This means the density changes from all agents broadcast to no agent sends a message. Using the proposed divergence measure function, this will result in a number representing a high degree of self-organization.

Self-integration: Unfortunately the paper does not describe if and how agents can discover other agents, or integrate a new agent into the existing system. The proposed logic would allow to add new agents following the "contains more elements" rule. In the beginning the solution candidate of the new agent is empty. When the new agent receives a message, the solution candidate has size $1 + n > 0$ (empty size), causing the agent updating its solution candidate and notify all other neighbors with his solution candidate containing his selection. When the neighbor agent receives that message, the size of the solution candidate will be $2 + n > 1 + n$ causing the neighbor propagating the new solution candidate to its neighbors. In conclusion the specification in the paper is not concrete about self-integration aspects. The simulation also does not need self-integration. It was executed with a fixed amount of agents without a not responding nor a new added agent. In the real world a self-integration mechanism is mandatory when building a new house, or something breaks and the house is not reachable anymore.

Self-healing: The paper does not discuss any self-healing mechanisms and there is no self-healing mechanism during a period. After each period the solution candidates in all agents get cleared resulting in a self-healing mechanism over periods. But if an agent fails, the system will not heal itself during that period and will find a solution assuming the failed agent participates. In the real world this would result in a minimal not optimal solution, which could also happen by not finding an optimal solution and exceeding the period.

Self-protecting: Assuming that reducing the load peaks protects the power grid we could talk about a self-protecting system. But there is no mechanism to defend the system against attacks from outside nor large-scale cascading failures beside the self-healing mechanism.

Self-stabilizing: When the system finds an optimal solution, the agents stop communicating until the next period starts which is after the definition of [2] a stable system.

Self-improving: Each agent works by always improving the previous taken decision. An agent makes a decision about his next schedule. When he perceives a new message he always searches for a better solution using the latest perceived information and corrects his previously made decision. This is self-improving even it does not use artificial machine learning techniques.

Self-explaining: The observing agent monitors all the communication in the system. This monitoring data could be used to explain how decisions were taken but why a specific solution was taken is hidden in the internals of each agent. The system is also human controllable and terminatable using the controller agent. In the real world this would be a big security vulnerability. Once started the agents should self-configure and each agent should be terminatable individually, but it is a big security vulnerability to allow termination of the whole system. This could potentially result in blackouts or worse things causing unrepairable damage.

Using the definition of [2] about the self-* properties to form a system in the domain of OC, most self-* properties are completely fulfilled or partly fulfilled. For a usage in the real world, the self-protection and self-integration aspects must be improved. Having no self-protection and self-integration mechanisms is probably caused by being a system only used in a simulation with a fixed amount of agents and no disturbances from outside. But even these two properties are not fulfilled, the system can be categorized as a system in the domain of OC.

Also the general concept of shifting demand and supply by changing the peoples behavior and paying a behavioral adaption cost is hard to imagine in the real world. This would be not only a technical challenge, but also a social one.

Conclusion

In this paper, we discussed whether the system used in "Distributed multi-objective scheduling of power consumption for smart buildings" (Marvin Nebel-Wenner et al. 2019) [1] fulfills the self-* properties defined in [2] to categorize it as a system in the domain of OC, or not.

The system can be categorized as a system in the domain of OC. It fulfills most self-properties completely or partly. For a usage in the real word, a concept for self-configuration and self-protection is needed, but in general robustness, adaptivity and manageability is given.

References

[1] M. Nebel-Wenner, C. Reinhold, F. Wille, A. Nieße, and M. Sonnenschein, "Distributed multi-objective scheduling of power consumption for smart buildings," Energy Inform, vol. 2, no. S1, p. 28, Sep. 2019, doi: 10.1186/s42162-019-0080-4.

[2] S. Tomforde, B. Sick, and C. Müller-Schloer, "Organic Computing in the Spotlight." arXiv, Jan. 27, 2017. Accessed: May 26, 2023. [Online]. Available: http://arxiv.org/abs/1701.08125

[3] S. Tomforde, J. Kantert, and B. Sick, "Measuring Self-organisation at Runtime - A Quantification Method based on Divergence Measures:," in Proceedings of the 9th International Conference on Agents and Artificial Intelligence, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 96–106. doi: 10.5220/0006240400960106.

[4] C. Hinrichs and M. Sonnenschein, "A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents," IJBIC, vol. 10, no. 2, p. 69, 2017, doi: 10.1504/IJBIC.2017.085895.