A Measurement Framework at Global and Local Levels for Hybrid Organic Computing Systems

Jonas Lange[1], Pia Schweizer[2], Elia Henrichs[2], Luna Kaendler[1], Sven Tomforde[1], and Christian Krupitzer[2]

[1] Intelligent Systems Group, Kiel University
{jla, st}@informatik.uni-kiel.de
[2] Department of Food Informatics & Computational Science Hub, University of Hohenheim
{pia.schweizer, elia.henrichs, christian.krupitzer}@uni-hohenheim.de

Abstract. Organic Computing (OC) systems adapt to changes in the system or environment to maintain system performance. Therefore, those systems integrate a control part, which can be implemented centralized (globally) or decentralized (locally)—mixed (hybrid) approaches are feasible. When determining the performance of such a system, various metrics can be observed, which generally cluster in system-specific and adaptation metrics. Whereas the first category is domain-specific and measures the system's performance, the second measures the adaptation performance and can be used system-independently for evaluating OC systems. In this paper, we present a measurement framework that takes into account this split in quality attributes. We show how to apply the measurement framework in a hybrid OC system based on the example of platooning coordination.

Keywords: Organic Computing · Measurement Framework · Performance Metrics · Platooning · Coordination · Platoon Formation

1 Introduction

Organic Computing (OC) [16] is a field that focuses on designing self-organizing, adaptive, and robust systems inspired by biological principles to autonomously manage their behaviour in dynamic environments. Therefore, these systems have two main parts: the system itself, called the System under Observation and Control (SuOC), and the adaptation part. The adaptation part usually follows the Observer-Controller Architecture [16], a framework that ensures the adaptability and self-management of systems. The Observer continuously monitors the system and its environment to collect data and identify deviations or patterns, while the Controller uses this information to make decisions and adjust the system's behavior to meet goals or maintain optimal performance. The Observer-Controller elements might be concentrated into a centralized system part (global control) or decentralized in the different sub-systems (local control). Also, mixed (hybrid) architectures are feasible.

However, the question arises of how differences between the architectures can be assessed and what an integrated measurement and evaluation approach for OC systems might look like. In this paper, we present a measurement framework to compare the quality of adaptation for different control architectures. As a complete discussion of the metrics for all aspects of adaptation quality is out of the scope of this work, we focus on relevant metrics for distributed systems, such as trust, benefits, costs, and emergence. We discuss the framework's application by conducting a simulation-based systematic evaluation of available global and local metrics within a platooning system. Platooning is the context-related coordination of vehicles to convoys in order to primarily exploit safety, capacity, and slipstream effects. The coordination of platoons can be centralized (i.e., external infrastructure services), decentralized (i.e., between the participating vehicles themselves), and hybrid as a combination. The basic problem of coordination decision-making is highly dynamic and, with this dynamic as well as the combination of (partially) autonomous local units and a system-wide objective function, it raises fundamental questions of OC [16]. For platooning as an OC example, the same challenges apply as for other use cases: As platooning is a multi-objective problem [20], there is no single objective function that could be optimized directly. This problem is also visible for the generic class of Self-Adaptive and Self-Organizing (SASO) systems, for which OC systems are one example. Corresponding efforts can be mapped to the overarching question of how the potentials and limitations of SASO systems can be made measurable and assessable, especially in comparison to conventional, purely centrally coordinated approaches. As both centralized and decentralized SASO systems come with challenges in cooperative multi-agent scenarios, the underlying project aims to build a hybrid SASO system, combining global optimization with autonomous subsystems [18]. Therefore, a top-down and a bottom-up approach converge towards the hybrid SASO system. Hence, this work focuses on examining the differences between centralized and decentralized control architectures.

The remainder of this work is structured as follows: Section 2 introduces the measurement framework. Section 3 describes the simulation environment and the general experimental setup for the platooning application. Section 4 presents the results of applying the relevant metrics from the measurement framework to the platooning application. Finally, Section 5 closes this paper.

2 Measurement Framework

The demands of SASO systems are diverse, extending beyond safety concerns to include the quality of their adaptation outcomes and the overhead required to make the adaptation decisions [6,2]. The following outlines a selection of metrics suited to evaluate the key properties of trust in Section 2.1, benefit and cost in Section 2.2, as well as emergence in Section 2.3. While metrics evaluating trust and emergence can be applied use-case independent, metrics evaluating the benefits and cost of a system are mostly use-case specific and require an explicit transformation to the particular application scenario. We also distinguish whether metrics are relevant at the global, local, or both levels simultaneously.

2.1 Trust

Trust in a system builds on a reliable, robust, and resilient performance. This includes consistently fulfilling its intended functions in stable and changing environments, even if unexpected changes occur (e.g., [15]). In particular, Stability, Robustness, and Unavailability are metrics that support the assessment of these three attributes and enable a comparison of trustworthiness between different systems. The three metrics are explained in the following.

Stability At the local level, the Stability metric focuses on the total time an agent spends in a desired state. Thus, the longer the total time, that the agents spent in the desired state, the more stable a system is considered. In [8], the stability of self-adaptation processes is assessed as how consistently a system selects expected configurations over time. Depending on the system's implementation, this can be on a global or local level. A system is stable if it frequently chooses high-probability configurations, reflecting normal behavior. Instability occurs when low-probability configurations are repeatedly chosen, signalling disturbances or failures. To detect undesired behavior in SASO systems, the authors use measures proposed by Kinoshita, that is, an activity factor and the fluctuation variance of the activity factor [12]. Generally, high stability is desirable because it indicates that the system maintains its structure and functionality consistently over time, showcasing greater reliability. However, frequent recalibrations could be favorable if they lead to a higher-quality outcome.

Robustness reflects the ability of the system to maintain a stable behavior when faced with unpredictable changes. It can be measured by the system's ability to maintain functionality during perturbations, with minimal variation in solution quality, by the new state of the system being close to the previous state, or by minimal changes within the system between the state during the perturbation and the new stable state [10]. In [11], the authors distinguish between a system's robustness under attack and long-term robustness. If a system never drops below a pre-defined baseline utility, it is considered to exhibit robustness.

Unavailability measures the time during which the system is not operational or unable to meet the required functionality. Following the work of [3], Unavailability U, i.e., the downtime of a system, can be derived as

$$U = \frac{MTTR}{MTTF + MTTR} \tag{1}$$

with $MTTR$ representing the *mean time to recover* and $MTTF$ being the *mean time to fail*. For systems with a fully centralized level of control, which represents a potential single point of failure, this metric is crucial and serves as a measure of the system's reliability.

2.2 Benefit and Cost

Achieving high-quality outcomes typically involves higher processing time or resource use, requiring trade-offs between benefit and cost. While understanding benefits aids in assessing costs, systems with strict time constraints must balance optimality and feasibility [23]. In the following, we consider two metrics for each attribute. We assess the benefit by applying the Situation Performance and Fairness, while the Latency and Overhead metrics serve to evaluate the cost.

Situation Performance SP serves as a measure of the quality of the final adaptation outcome and evaluates whether and how effectively the system fulfills its intended purpose. The higher the SP, the higher the quality or benefit, the lower the connected cost. In [22], the authors derive the Situation Performance of a system by comparing the actual cost for an adaptation decision C_{subsit} to the maximum possible cost $C_{\max}$. Therefore, they divide a situation sit into numerous sub-situations $subsit$.

$$SP = 1 - \frac{\sum_{\text{subsit} \in \text{sit}} C_{\text{subsit}}}{\sum_{\text{subsit} \in \text{sit}} C_{\max}} \tag{2}$$

The cost thereby represents a use-case-specific measure that requires an appropriate transfer depending on the system's purpose.

Fairness can be categorized into various dimensions, such as resource fairness, i.e., equally distributed access to resources, benefit fairness, or responsibility fairness. Focusing on the example of benefit fairness, the metric evaluates how profits are distributed among agents. The authors of [13] determine the fairness among all individuals n by determining the Gini coefficient G.

$$G = \frac{2 \sum_{i=1}^{n} i x(i)}{n \sum_{i=1}^{n} n x(i)} - \frac{n+1}{n} \tag{3}$$

Here, n refers to the total number of individuals in the group being analyzed, composed of single individuals i, and their corresponding values $x(i)$, representing a measurable attribute. A G of 0 indicates maximum fairness among individuals, where no single agent or group of agents consistently gains or suffers disproportionately, while an index of 1 refers to minimum fairness.

Latency generally refers to the delay in system response. It reflects a time cost associated with the system's decision-making and adaptation processes when encountering disturbances. Latency L can be measured by comparing the time the system requires to adapt to changing environments T_{change} to the time it takes to perform its usual functionality without disturbances T_{usual} [10].

$$L = \frac{T_{change}}{T_{usual}} \tag{4}$$

A lower L indicates that the system is able to react more quickly to changes, enhancing responsiveness. However, a higher L may suggest a more detailed search for optimal solutions, which, while introducing delays, can lead to a higher quality of the final adaptation outcome, but may also result in conflicts with time constraints. This trade-off between speed and quality is a key consideration in systems with time-sensitive constraints.

Overhead For adaptation decisions to take place, information about the participating agents is required. The gathering and processing of information represents, among others, a form of overhead, which grows as the number of agents increases or as more detailed information about each agent is required. In general, overhead creates cost, but in turn, is a prerequisite for benefits. The authors of [21] further distinguish between communication, computation, memory, and monitoring overhead.

2.3 Emergence

Emergence describes a phenomenon where systems evolve from chaotic conditions to higher-order levels without being explicitly programmed beforehand. It arises from the interaction of many individuals who operate without central control [7]. Different ways to detect and measure emergence are presented below, comprising the Interaction, Entropy, and Oscillation Detection metrics.

Interaction can be assessed by counting the number of effective interactions I_t that take place for every individual i at a given time t [4].

$$I_t = \sum_{\forall i} \delta_{i,t} \tag{5}$$

$\delta_{i,t}$ changes from 0 to 1 for an effective interaction, which means that the interaction resulted in a state change. An emergent behavior can thereby be detected if the results over time deviate from normality.

Entropy reflects the degree of disorder. Among various types of entropy, Shannon's Entropy is a prominent example in the field of information theory and enables the evaluation of a system's information content [5]. For Shannon's Entropy SE, the possible system states x of a system X follow a probability distribution $P(x)$. SE can be calculated using Equation (6).

$$SE = -\sum_{x \in X} P(x) \log P(x) \tag{6}$$

A system with a low entropy represents a high certainty regarding the probability of specific system states to occur; thus, the information content of the system is low. Conversely, a high entropy indicates a high information content as the probability of a specific system state occurring is low, resulting in a high uncertainty regarding predictions about system states.

Oscillation Detection OD represents the interval k after which a system state S_t at time t has already occurred before [2], . The degree of similarity between system states necessary to result in a detected reoccurrence depends on the use-case and the definition of stability.

$$OD = \begin{cases} k & \text{if } S_t \leftrightarrow S_{t-k} | k \geq 1, t \geq 0 \\ 0 & \text{else} \end{cases} \tag{7}$$

It equals zero if there was no repeated state observable and otherwise results in the respective time interval in between the states' occurrence. The repeated occurrence of an OD of 1 describes a steady state. Transferring this to the local system layer, one could study the frequency with which agents change their state and identify persistent fluctuations where the agents struggle to settle into a stable configuration.

2.4 Further Metrics

The metrics presented here provide only a partial view, which is far from exhaustive. Other relevant literature providing further metrics includes the work of Kaddoum et al. [10]. The authors propose several measures for evaluating the adaptive properties of self–*systems, introducing metrics that focus on methodological, architectural, intrinsic, and runtime evaluation criteria. Eberhardinger et al. [6] focus on the key metrics for assessing the performance of self-organization algorithms, with particular emphasis on time and solution quality. Furthermore, Birdsey et al. [2] present a compilation of metrics that enable the evaluation of the two properties, self-adaptation and self-organization, in isolation.

3 Experimental Setup

As for evaluating the performance of SASO and OC systems, adaptation metrics and system-specific metrics must be considered. The experiments aim to integrate domain-specific use case metrics and to show differences between centralized and decentralized control architectures. The experiments were conducted using the example of platoon coordination, which describes a highly dynamic multi-agent environment. For simulating the vehicles and traffic, we rely on the open-source SUMO (Simulation of Urban MObility, [1]) simulator. For simulating the platooning functionality, we use the Python API of Plexe [19], an open-source SUMO extension. In the following, Section 3.1 explains the applied traffic situations. Section 3.2 illustrates the implementation of two platooning algorithms, one with a centralized and one with a decentralized control architecture.

3.1 Traffic Scenarios

Some parameters were set statically to keep the number of variable traffic scenarios in check.

Table 1. Platooning parameters

Parameter	Value
Platooning spacing	5 Meters
Vehicle headway	1.5 Seconds
Max. platoon size	5 Vehicles

Table 2. Vehicle parameters

Parameter	Value
Min. speed	80 km/h
Max. speed	160 km/h
Vehicle length	4.3 Meters

Table 1 shows the parameters for the platoons, while Table 2 shows the static parameters for the individual vehicles. For simplicity, all vehicles are of the same dimensions and equipped with the same engines, only their desired velocity differs, which is set to a random value between 80 and 160 km/h. We test three different traffic densities on three maps with three different platooning participation rates. Hence, we evaluate the two platooning algorithms on 27 different traffic scenarios. Each scenario simulates exactly 65 minutes of traffic, of which the first 5 minutes are not considered for evaluation since the environment is set up with vehicles on an empty highway.

Each of the three maps represents a unique challenge that can occur on highways. For all scenarios, the cars spawn at the beginning of a one-kilometer-long startup section and despawn when they reach the end of the one-kilometer-long cool-down section. Yet only the 10 km long main road section is evaluated, as spawning and de-spawning vehicles (on the startup- and cool-down sections) might disrupt the natural traffic flow. In the *Straight* map, the cars drive on a straight, three-lane highway. The second map is the *Lane Reduction* map, which starts with a straight highway, but at 5 km, the rightmost lane ends; thus, the 4-lane highway turns into a 3-lane highway. This leads to an increased traffic density from that point onward. The third map is called the *Y-map*, which starts with a straight section. After 5 km, the road splits into two, resembling the letter "y" turned by 90°. This introduces a novel challenge to platooning, as platoons might need to split up to ensure that each vehicle reaches its destination.

Traffic Density will likely influence the performance of platooning coordination systems. We therefore consider the following scenarios: Low Density, Medium Density, and High Density. Traffic density should not be too low, resulting in high inter-vehicle distances and no opportunity for platoons to form. Traffic density should also not be too high, resulting in a traffic jam. The High Density is set to 5400 vehicles per hour (v/h) as this is estimated to be the maximum capacity of a three-lane highway [17]. The Low Density was set to 1200 v/h to ensure that, on average, there are 10 vehicles within a 1 km stretch of highway. The Medium Density is set to 3300 which is the average of Low Density and High Density.

Platooning Participation is seen as voluntary for vehicles. Hence, we consider that half of the vehicles (50%) want to join platoons. For comparison, we also include scenarios without (0%) and with forced platooning (100%).

3.2 Platoon Coordination

As a general architecture for platoon coordination, we consider two possibilities. Nearly all proposed platooning coordination systems follow one of these two architectures [14]. In a **centralized** architecture, a central control unit is in place to make platooning decisions. In a **decentralized** architecture, individual vehicles coordinate the formation of platoons. In real-world applications, the feasibility of either of these architectures depends on the infrastructure in place (e.g., communication infrastructure). In [18], we further discuss the advantages and disadvantages of the two architectures. As the inter-vehicle communication within platoons (e.g., keeping correct inter-vehicle gap, platoons changing lanes, etc.) is handled by Plexe, the centralized and decentralized platoon coordination algorithms are mainly concerned with assigning vehicles to platoons. Figure 1 shows the three main states of the platooning coordination task.

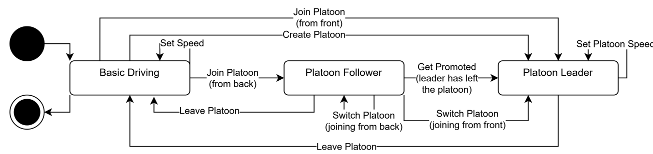

Fig. 1. State machine for the platooning coordination task from the perspective of an individual vehicle. Vehicles can join, leave, and switch platoons. Vehicles not in a platoon control their own speed and lane, while platoon leaders set these for the group.

The authors of [9] propose two static platoon coordination algorithms. The proposed centralized and decentralized algorithms have only three parameters (α, r, and m). This fits well into the experimental design because it focuses on the metrics in relation to the traffic scenarios rather than optimizing algorithm parameters. In assigning vehicles to platoons, the algorithms try to minimize two values: One is the physical distance between the candidate vehicles c and the target platoons' t current position $d_p(c,t)$. The other is their difference in speed $d_s(c,t)$. These two values are weighted against each other using the parameter α, which we set to its default value $\alpha = 0.5$:

$$f(c,t) = \alpha \cdot d_s(c,t) + (1-\alpha) \cdot d_p(c,t) \tag{8}$$

In the decentralized algorithm, each candidate vehicle applies Equation (8) individually to find the best target platoon inside its search radius $r = 500$ m. In addition, platoons with a traveling speed outside of the maximum speed deviation $m = 0.2$ are not considered (e.g., a vehicle with a desired speed of 100 km/h would only consider platoons traveling at speeds 80-120 km/h). Vehicle c then selects the candidate platoon t, which minimizes $f(c,t)$. If there are no platoons within r and m, the vehicle creates a new platoon (only containing itself). If two vehicles want to join the same platoon, first-come, first-served is applied.

In the centralized algorithm, a central controller first collects all possible assignments $(c_i, t_j, f(c_i, t_j))$ of candidate vehicles c_i to target platoons t_j. Like the decentralized algorithm, possible assignments outside the search range $r =$ 500 m, and the maximum speed deviation $m = 0.2$ are not considered. The central controller then assigns each vehicle to the corresponding optimal platoon. The optimal platoon t_i for a candidate vehicle c is the one that minimizes $f(c, t_i)$. During the assignment, the centralized controller applies a greedy approach. It finds the best target platoon for each candidate vehicle in random order while removing possible assignments $(c_i, t_j, f(c_i, t_j))$ if a vehicle has already been assigned to platoon t_j in this iteration.

We adopted and adapted these two algorithms for our experiments. Our major changes to the original algorithms are twofold: First, in the original implementation [9], vehicles were only searching for platoons in front of them to join them from behind. In our implementation, vehicles search for platoons in front and behind and can join platoons from the back or front. This effectively increases the number of joinable platoons for each vehicle and, thus, the likelihood of finding better-matching platoons. Second, in contrast to the original implementation, our variant allows vehicles to change platoons.

4 Results and Discussion

To quantify the potentials and limitations of SASO systems, we derived use case-specific measures from the metrics of Section 2. We focused on those metrics that specifically highlight the differences between the centralized and decentralized algorithms, providing insights into their respective strengths and limitations.

4.1 Trust

The *Robustness* metric measures the ability of a system to maintain functionality during perturbations. While there are no sudden and unexpected perturbations in the evaluated traffic scenarios, there are static perturbations, namely the lane reduction and the y-split. At these points, the systems must adapt as the environment suddenly changes. The platoon coordination system aims to assign as many vehicles to adequate platoons as possible. Therefore, we interpret the functionality of the system to be the number of vehicles driving in platoons.

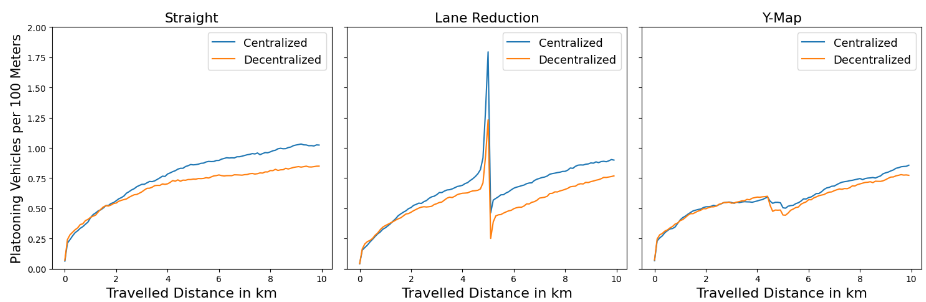

Fig. 2. Robustness as the number of vehicles in a platoon over the length of the map. The scenarios selected are with medium traffic and 50% platooning participation.

Figure 2 shows the number of vehicles driving in platoon formations at different points of the road. The straight map can be seen as the baseline, as there are no disturbances on the road. Here, the number of platooning vehicles per 100 m rises steadily for both the centralized and the decentralized algorithms. However, the centralized algorithm, in general, has a higher platooning density. This general performance difference is relevant to consider when evaluating the recovery after the disturbances for the Robustness metric because the focus here is on robustness, not on general performance. On the lane reduction map, the centralized and decentralized algorithms start out similarly. Just before the 5 km mark, the number of platooning vehicles suddenly spikes. This is due to a general high traffic density before the lane reduction point, where some vehicles need to merge, causing a slowdown of the following traffic. The main difference is visible after the lane reduction; here, the centralized system recovered to a higher platooning density than the decentralized system. But when compared to the straight scenario, this difference can be attributed to the better general performance of the centralized system. Thus, for the lane reduction scenario, the two systems are similarly robust. Finally, on the y-map, both systems perform similarly. The functionality drops slightly at the y-split as some vehicles split up from their current platoons. Here, the difference between the centralized and decentralized systems is much smaller than in the straight scenario. This indicates that the decentralized approach, while being, in general, less effective, is more robust in this scenario.

Robustness reflects a system's ability to recover and return to a stable or desired state after a disturbance, ensuring long-term operational continuity. The proposed metric allowed us to assess system robustness under different types of disturbances. The results showed that the robustness varies depending on the disturbance type, revealing potential weak points that could ultimately reduce trust in the system. Explicitly identifying these vulnerabilities enables targeted improvements to enhance system resilience. Therefore, we consider this metric highly relevant within our framework, particularly in the *Trust* category.

4.2 Benefit and Cost

The *Situation Performance* describes the quality of the final adaptation outcome. Concerning the use case, one of the main purposes favoring platooning is the reduction of fuel consumption [24]. Thus, we assume a higher Situation Performance if all vehicles that had the intention to participate in platooning achieve fuel savings compared to a scenario without platooning, where the SASO system was not active. Therefore, we calculate the individual Situation Performance using Equation (2), and consider the mean fuel consumption of a vehicle that wanted to platoon as C_{subsit}, while the maximum cost C_{max} represents the mean fuel consumption of the same vehicle in the scenario without platooning. This follows the assumption that platooning leads to fuel savings. If this assumption does not hold, the Situation Performance yields a negative value.

Figure 3 illustrates the resulting Situation Performance of both centralized and decentralized systems operating in scenarios with 50% and 100% platooning participation, respectively, depending on the traffic density in vehicles spawning per hour. Here, we chose to focus on the results of the lane reduction map as the static disturbance of the road bottleneck led to traffic congestion for high traffic densities, which represents a disturbance. The road split in the Y-map also represents a disturbance, but did not result in congestion. Despite the challenge, the results show that platooning—regardless of traffic density—yielded a higher Situation Performance and thus improved fuel consumption. At a low and medium traffic density, a higher participation rate led to a higher Situation Performance, and the decentralized system outperformed the centralized one. Conversely, with high traffic density, the trend reversed, making a lower participation rate favorable, as those scenarios resulted in a higher Situation

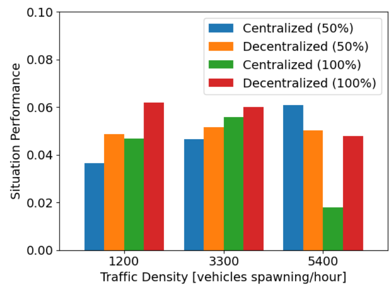

Fig. 3. Situation Performance of platooning vehicles at different platooning participation rates, depending on the traffic density on the lane reduction map.

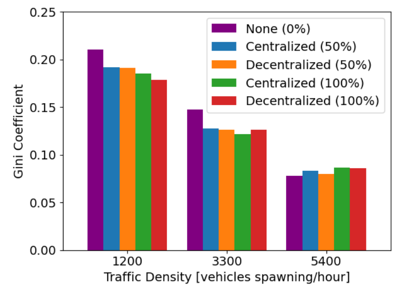

Fig. 4. Gini Coefficient regarding the fuel consumption of all vehicles at different platooning participation rates, depending on the traffic density on the lane reduction map.

Performance. It additionally came to a switch in performance for the centralized and decentralized approaches at the medium platooning participation rate. The results indicate that high platooning participation especially benefits low and medium traffic densities, as it increases the number of potential platooning partners, enhancing the opportunity to save fuel. With a dense distribution of platoons on a crowded road segment, the inflexibility of all road users increases, reducing the positive effects of platooning and, thus, the Situation Performance. These observations indicate that with the current approaches in place, too-high traffic counteracts the effect of platooning while emphasizing the presence of a sweet spot marking the optimal fuel savings at a specific traffic density and participation rate.

On a local level, one can assess the *Fairness* among the participating agents with respect to their fuel consumption. This fairness, expressed as the Gini coefficient (Eq. (3)), is presented in Figure 4 depending on the traffic density in vehicles spawning per hour. For space reasons, we again focus on the results of the lane reduction map, without platooning (0% platooning participation) as well as 50% and 100% platooning participation with a centralized or decentralized coordination algorithm. The results show that a higher vehicle density resulted in a higher equality among the agents, with the Gini coefficient getting closer to a value of zero. While at low and medium traffic levels, a higher participation rate led to greater equality, at high traffic, higher participation resulted in decreased equality. This shows that an increasing vehicle load increases fairness, as fewer vehicles profit from platooning.

The benefit of improved fuel efficiency when platooning comes with necessary pre-investments in the form of, among others, time cost. Therefore, we determined the *Latency* of both approaches, centralized and decentralized, at a maximum platooning participation rate, as here, we assumed the highest computational effort. The lane reduction and y-split seen as a disturbance resulted in no significant difference between the centralized and decentralized approaches when taking the straight scenario as a baseline. However, a larger impact on both system's Latency proved to have the traffic density. With higher throughput, the delay of both systems increased, with the decentralized system having a maximum Latency of 5.25 and, thus, less than half that of the central system with 13.65. This significant difference can be explained by the fact that the centralized approach always tries to find a solution for every vehicle at once, whereas the decentralized approach considers each vehicle individually, resulting in a lower calculation effort.

While the Situation Performance metric provided valuable insights regarding the quality of the final adaptation outcome on a global level, the Fairness metric enabled the performance assessment on a local level. A benefit on a global level is necessary to reason for the system as a whole, whereas understanding the distribution of those benefits on a local level helps identify potential dissatisfaction or even agent withdrawal from participation. Therefore, independent of the SASO system's architectural approach, both perspectives must be considered, making these metrics crucial to our measurement framework. Closely linked to the benefits are the associated costs. The Latency metric, which captures cost in the form of delay, allowed us to compare centralized and decentralized approaches in terms of the time required for adaptation. A key question is whether these incurred costs are justified by the benefits gained, highlighting an inherent trade-off, one that is evaluated differently depending on whether the perspective is global or local. Furthermore, although not explicitly evaluated in this work, assessing the overhead in terms of message exchanges could be a valuable metric for future research. It may provide insights into local-level costs, particularly enabling the evaluation of the effort required for an agent to participate.

4.3 Emergence

Here, we apply the *Interaction* metric, which counts the number of effective interactions between vehicles. The vehicles are constantly communicating in order to advertise and find platoon opportunities. An effective interaction takes place if a state change occurs. In the platooning scenario, a state change equates to a change in driving strategies for at least one of the vehicles involved. A strategy change happens, for example, when a vehicle joins a platoon, switches platoons, or the vehicles' platoon changes lanes. Figure 5 shows how many effective interactions took place mapped over the length of the highway. We selected the

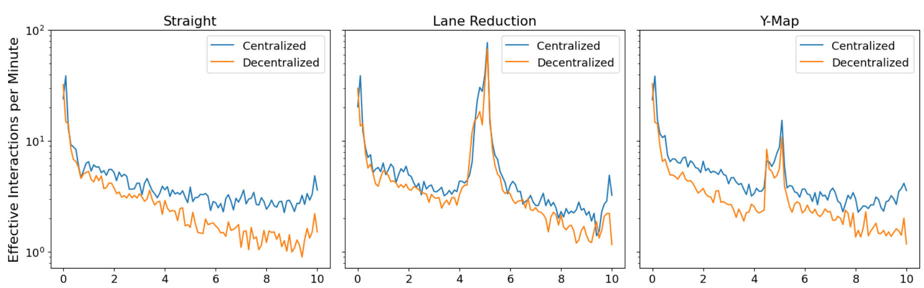

Fig. 5. Interaction as vehicles' strategy changes over the length of the map. The scenario is considering 100% platooning participation at a high traffic density.

scenarios with high traffic density and 100% platooning participation because the differences between the centralized and decentralized systems are most clear. In scenarios with lower platooning participation and/or lower traffic density, the trends described below are similar, but the differences between the algorithms are less pronounced. On all three maps, the number of interactions spikes initially as vehicles immediately try to find available platoons and join them. The system then stabilizes as vehicles find adequate platoons and state-changing interactions decrease. In the lane reduction map and the y-map, the number of effective interactions then spikes around the traffic obstacle (at 5 km), as the vehicles need to reorganize to navigate the obstacle. In general, the centralized controller seems to facilitate more effective interactions. This could lead to a better overall performance but also to a higher overhead.

The results of the Interaction metric indicate an emergent behavior of both approaches, centralized and decentralized, when participants encounter disturbances, forcing them to reorganize. We found that out of the three proposed metrics for emergence, the interaction metric is most applicable to our application. Yet the Entropy and Oscillation Detection metrics, might be more suitable for future application scenarios. Hence, we considered all three metrics within our proposed framework for an extensive assessment of a SASO system's emergence: the Interaction metric—to detect emergence, Entropy—to evaluate the degree of (dis)order, and the Oscillation Detection metric—to measure fluctuations and, consequently, the stability of the established order.

4.4 Discussion

Our results have shown that the centralized and decentralized algorithms perform differently in different situations and relative to different metrics. The *Interaction* metric shows that the centralized algorithm facilitates more effective interactions due to its globalized view. On the upside, this leads to a generally higher platooning density (as shown by the *Robustness* metric). On the downside, the *Latency* metric shows that this globalized view also requires more resources, especially when the traffic density increases. When measuring the benefit of applying the centralized vs. the decentralized algorithm, the *Situation Performance* metric shows that there is no clear winner, and the performance is situation-dependent. In some scenarios, the higher general platooning density leads the centralized system to perform better. In other scenarios, savings in overhead and general robustness lead the decentralized system to perform better.

The results show that rather than applying static algorithms, it is necessary to apply intelligent systems that learn to recognize different situations and act accordingly. The decentralized algorithm could be improved by applying Reinforcement Learning. While the centralized algorithm could benefit from optimization techniques that improve the coordination. The results also reinforce the necessity for a hybrid system [18], which combines the strengths of centralized and decentralized systems. Our proposed measurement framework aims at a comprehensive evaluation of SASO systems. We identified three key metric categories: *Trust*, *Benefit and Cost*, and *Emergence*. Our findings demonstrate that no single metric can fully assess an entire category; instead, a combination of metrics is required to capture a system's strengths and weaknesses. Moreover, evaluating both the global and local levels is essential, as their distinct interests influence the system as a whole. Different metrics are needed for each level to ensure a well-rounded assessment. By incorporating both perspectives, our framework not only provides a holistic evaluation but also enables a direct comparison of different approaches. We see two main limitations of this work. First, the proposed measurement framework is far from exhaustive. While this work aims to include the most relevant metrics, we recognize that some excluded ones could round off the framework. Second, with the platooning application, we aim to highlight the ability of the measurement framework to showcase the differences between centralized and decentralized architectures. Yet, one application scenario is not sufficient to show the general applicability of the proposed measurement framework.

5 Conclusion

When it comes to measuring the effects of SASO and OC systems, there is rarely one singular metric that is able to fully capture a system's complexity and overall impact. In this work, we propose a measurement framework to assess such systems' dynamics. Our framework provides diverse measures from the areas of *Trust*, *Benefit and Cost*, as well as *Emergence*. We evaluated the measurement framework on the example of platooning, applying two coordination strategies—centralized and decentralized. None of the two approaches was preferable over all scenarios and the performance was situation-dependent. This work has shown that the framework, while not exhaustive, is able to highlight the strengths and weaknesses of centralized and decentralized systems. Thus, it is in a good position to be useful in evaluating hybrid approaches, which is the main objective of our future work.

Acknowledgments

This work is part of the InHOSaS project which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—516601628.

References

1. Alvarez Lopez, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: IEEE ITSC. pp. 2575–2582 (2018)
2. Birdsey, L., Szabo, C., Falkner, K.: Identifying self-organization and adaptability in complex adaptive systems. In: SASO. pp. 131–140 (2017)
3. Candea, G., Cutler, J., Fox, A.: Improving availability with recursive microreboots: A soft-state system case study. Performance Evaluation **56**(1-4), 213–248 (2004)

4. Chan, W.K.V.: Interaction metric of emergent behaviors in agent-based simulation. In: WSC. pp. 357–368 (2011)
5. Crutchfield, J.P., Young, K.: Inferring statistical complexity. Phys. Rev. Lett. **63**, 105–108 (Jul 1989)
6. Eberhardinger, B., Anders, G., Seebach, H., Siefert, F., Reif, W.: A research overview and evaluation of performance metrics for self-organization algorithms. In: SASO Workshops. pp. 122–127 (2015)
7. Goldstein, J.: Emergence as a construct: History and issues. Emergence **1**(1), 49–72 (1999)
8. Goller, M., Tomforde, S.: On the stability of (self-) adaptive behaviour in continuously changing environments: A quantification approach. Array **11**, 100069 (2021)
9. Heinovski, J., Dressler, F.: Where to decide? centralized versus distributed vehicle assignment for platoon formation. T-ITS (2024)
10. Kaddoum, E., Raibulet, C., Georgé, J.P., Picard, G., Gleizes, M.P.: Criteria for the evaluation of self-* systems. In: ICSE SEAMS. p. 29–38. SEAMS, Association for Computing Machinery, New York, NY, USA (2010)
11. Kantert, J., Tomforde, S., Müller-Schloer, C., Edenhofer, S., Sick, B.: Quantitative robustness – a generalised approach to compare the impact of disturbances in self-organising systems. In: ICAART,. pp. 39–50. INSTICC, SciTePress (2017)
12. Kinoshita, T.: Basic characteristics of a macroscopic measure for detecting abnormal changes in a multiagent system. Sensors **15**(4), 9112–9135 (2015)
13. Kohler, T., Steghöfer, J.P., Busquets, D., Pitt, J.: The value of fairness: Trade-offs in repeated dynamic resource allocation. In: IEEE SASO. pp. 1–10. IEEE (2014)
14. Lesch, V., Breitbach, M., Segata, M., Becker, C., Kounev, S., Krupitzer, C.: An overview on approaches for coordination of platoons. T-ITS **23**(8), 10049–10065 (2022)
15. Moskowitz, F., McLean, J.B.: Some reliability aspects of systems design. IRE Transactions on Reliability and Quality Control **PGRQC-8**, 7–35 (1956)
16. Müller-Schloer, C., Tomforde, S.: Organic Computing - Technical Systems for Survival in the Real World. Birkhäuser (2017)
17. Ober-Sundermeier, A.: Entwicklung eines Verfahrens zur Stauprognose an Engpässen auf Autobahnen unter besonderer Berücksichtigung von Arbeitsstellen. Ph.D. thesis, University of Kassel (2003)
18. Schweizer, P., Lange, J., Henrichs, E., Tomforde, S., Krupitzer, C.: Towards a hybrid architecture for self-adaptive and self-organizing systems. In: IEEE ACSOS-C. pp. 41–46. IEEE (2024)
19. Segata, M., Lo Cigno, R., Hardes, T., Heinovski, J., Schettler, M., Bloessl, B., Sommer, C., Dressler, F.: Multi-Technology Cooperative Driving: An Analysis Based on PLEXE. TMC **22**(8), 4792–4806 (8 2023)
20. Sturm, T., Krupitzer, C., Segata, M., Becker, C.: A taxonomy of optimization factors for platooning. T-ITS **22**(10), 6097–6114 (2021)
21. Szabo, C., Sims, B., Mcatee, T., Lodge, R., Hunjet, R.: Self-adaptive software systems in contested and resource-constrained environments: Overview and challenges. IEEE Access **9**, 10711–10728 (2021)
22. Taranu, S., Tiemann, J.: On assessing self-adaptive systems. In: PERCOM Workshops. pp. 214–219 (2010)
23. Van Der Donckt, J., Weyns, D., Iftikhar, M.U., Buttar, S.S.: Effective decision making in self-adaptive systems using cost-benefit analysis at runtime and online learning of adaptation spaces. In: ENASE. pp. 373–403. Springer (2019)
24. Zabat, M., Stabile, N., Frascaroli, S., Browand, F.: Drag forces experienced by 2, 3 and 4-vehicle platoons at close spacings. SAE transactions pp. 1173–1181 (1995)