# Measuring Robustness in Hybrid Central/Self-Organising Multi-Agent Systems

Yaser Chaaban and Christian Müller-Schloer
Institute of Systems Engineering
Leibniz University of Hanover
Hanover, Germany
chaaban,cms@sra.uni-hannover.de

Jörg Hähner
Institute of Organic Computing
University of Augsburg
Augsburg, Germany
joerg.haehner@informatik.uni-augsburg.de

***Abstract*—It is noteworthy that the definition of system robustness varies according to the context in which the system is used. Therefore, manifold meanings of system robustness were introduced in literature. Additionally, various formal measures and metrics were presented to achieve the system robustness. In previous papers, we proposed a new concept to keep a multi-agent system robust when deviations from planned (desired) behaviour occur in the system. This concept introduces a robust hybrid central/self-organising multi-agent system. The scenario used in this work is a traffic intersection without traffic lights. In this paper, we extend our prototype implementation with the aim of making it capable of handling disturbances (accidents) occur in the system environment (intersection) aiming to completely realise our vision. Simultaneously, we develop an appropriate metric for the quantitative determination of the robustness.**

***Keywords-Robustness; Organic Computing; Hybrid Coordination; Multi-Agent Systems***

## I. INTRODUCTION

Organic Computing (OC) has the objective to use principles that are detected in natural systems. In this case, nature can be considered as a model aiming to cope with the increasing complexity of the recent technical systems [3]. Consequently, OC tries to develop systems that are adaptive, flexible and robust at the same time utilising advantage of the organic properties of OC. In this regard, the robustness of OC systems is a key property, because the environments of such systems are dynamic.

In organic systems, the design of the system architecture plays a main role in achieving a robust system so that its performance has to remain acceptable in the face of deviations or disturbances occurred in the system (intern) or in the environment (extern). That means, the developing of robust systems needs to take into account that degradation of the system's performance in the presence of such disturbances should be limited in order to maintain a satisfying performance. Therefore, a robust system has the capability to act satisfactorily even when conditions change from those taken into account in the system design phase. Nevertheless, this capability has to be retained, because of the increasing complexity of novel systems where the environments change dynamically. As a result, fragile systems may fail unexpectedly even due to slightest disturbances. Thus, a robust system will continue working in spite of the presence of disturbances by counteracting them with corrective interventions.

Considering the system design paradigm, it should be decided whether the system architecture will be centralised or decentralised. Centralised approach is the paradigm where the system is based on a centralised architecture (there is a central controller and the components of the system are not fully autonomous). On the other hand, decentralised approach means that the system has a distributed (there is no central controller and all components of the system are autonomous) or a hierarchical architecture (the components of the system are semi-autonomous in which they are locally centralised) [4]. Based on this, distribution possibilities of system architecture have important implications for system robustness.

Although the decentralised approach would have some advantages over the centralised one, especially scalability, the hybrid approach containing both centralised and decentralised at the same time is applicable and even may be much better than the use of each one separately. The hybrid approach should be robust enough against disturbances, because robustness is an indispensable property of novel systems. Additionally, it represents the interaction between decentralised mechanisms and centralised interventions. In other words, the hybrid approach exhibits the central/self-organising trait simultaneously. This means that a conflict between a central controller (e.g., a coordination algorithm) and the autonomy of system’s components should be solved in order to achieving the robustness of the system.

For this purpose, OC uses an observer/controller (o/c) architecture as an example in system design. Using the (o/c) design pattern proposed in [5], the behaviour of OC systems can be observed and controlled. A generic o/c architecture was presented in [6] to establish the controlled self-organisation in technical systems. This architecture is able to be applied to various application scenarios.

During the last years, the progress in communication and information technologies was significant. Consequently, a lot of investigations were done aiming to improve transport systems so that the “Intelligent Transportation System (ITS)” was developed. ITS have several applications in traffic and automotive engineering. According to ITS, numerous notions were distinguished such as, among other, intelligent vehicles, intelligent intersections, and autonomous vehicles. In this context, a traffic intersection without traffic lights can be considered as a main testbed to apply the hybrid approach, where autonomous agents are autonomous vehicles, and the controller of the intersection is the central unit.

## II. THE ORIGINAL SYSTEM

In previous papers, we introduced a system for coordinating vehicles at a traffic intersection using an o/c architecture [1][2]. The traffic intersection is regulated by a controller, instead of having physical traffic lights. Figure 1 shows a screenshot from our project. In this regard, we proposed a new multi-agent approach which deals with the problem occurring in the system wherever multiple agents (vehicles) move in a common environment (traffic intersection without traffic lights). We presented the desired system architecture together with the technique that is to be used to cope with this problem. This architecture was an o/c architecture adapted to the scenario of traffic intersection.

In both earlier papers, we implemented the generic o/c architecture adapted to our traffic scenario and accomplished our experiments assuming that no deviations from plan occur in the system. The evaluation of the concept was carried out based on the basic metrics: throughput, waiting time and response times [1] [2].

In this paper, we continue with the implementation of the case when disturbances (accidents) arise in the system (intersection) to completely realise our vision. Consequently, the system performance remains effective and will not deteriorate significantly or at least the system will not fail.

Additionally, an appropriate metric for the quantitative determination of the robustness will be developed and presented in this paper.

This paper is organised as follows. Section 2 describes our original system introduced in [1][2]. Section 3 presents a survey of related work concerning robust agent-based approaches used for fully autonomous vehicles within an intersection without traffic lights, in addition to various methods for measuring robustness. Section 4 is the main part of this paper. Firstly, it describes the interdisciplinary methodology, “Robust Multi-Agent System” (RobustMAS), developed in this paper. After that, it presents the measurement of robustness and gain according to the RobustMAS-concept. Section 5 introduces the evaluation of the system performance by means of experimental results. Section 6 draws the conclusion of this work. Finally, the future work is explicated in Section 7.

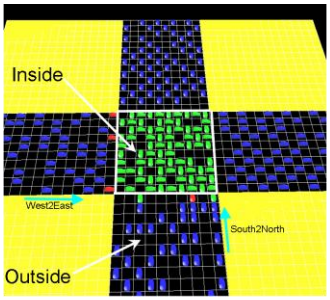

Figure 1. The traffic intersection without traffic lights

## III. STATE OF THE ART

Keeping a system robust in presence of disturbances or deviations from plan was investigated by researchers for years. Consequently, many approaches or architectures were introduced towards building robust systems.

In the literature, there are enormous works concerning safety properties of usual traffic intersections that concerns only human-operated vehicles. Additionally, there are some works in connection with safety measures of autonomous vehicles within an intersection. In this paper, we focus the discussion of related work on robust agent-based approaches used for fully autonomous vehicles within an intersection without traffic lights. Furthermore, we consider various methods for measuring robustness.

In this regard, according to our knowledge, there are no projects that focus on the robustness of autonomous vehicles within an intersection without traffic lights, where disturbances occur.

A study of the impact of a multi-agent intersection control protocol for fully autonomous vehicles on driver safety is presented in [7]. In this study, the simulations deal only with collisions in intersections of autonomous vehicles aiming to minimise the losses and to mitigate catastrophic events. However, it can be noted that the study has not considered the robustness of the intersection system.

### *A. Measures for robustness*

Many research projects deals with system robustness and they make an effort to measure the robustness and to find an appropriate metric for it. These projects are in various kinds of science. Robustness metrics play the role to mitigate the expected degradation of the system performance when disturbances occur.

There is a clear lack of study of these metrics in designing robust multi-agent systems. This paper raises the question: “How the robustness can be guaranteed and measured in technical systems?”.

In literature, there are diverse potential measures of system robustness. Every measure of robustness is based and designed according to the definition of the robustness concept in a specific context. The most common robustness measure uses the robustness definition related to definition of performance measure. Some robustness measures estimate the system performance using the average performance and its standard deviation, the signal-to-noise ratio, or the worst-case performance. Other robustness measures take into account the probability of failure of system as well as the maximum deviation from benchmark where the system has still the ability to deal with failures [8].

### *B. Generalised robustness metric*

Viable quantitative approaches in order to measure robustness are required. Some approaches were introduced, among others, in [9][10][11]. Both approaches; the FePIA procedure in [9] and the statistical approach in [10] are general approaches and consequently can be adapted to specific purposes (arbitrary environment). In both approaches, diverse general metrics were used to quantify robustness. This metrics estimate specific system features in the case of disturbances (perturbations) in components or in environment of the system. Additionally, this metrics were mathematically described. Both approaches in [9] and in [10] are applicable in embedded systems design [11] where embedded systems are designed as Systems on Chip (SOC). These both approaches do not comply with the RobustMAS-concept we use to characterise robustness.

To the best of our knowledge, this paper represents the first study towards measuring the robustness of hybrid central/self-organising multi-agent systems in intersections without traffic lights using the organic computing (OC) concept.

## IV. THE APPROACH

The Organic Computing initiative aims to build robust, flexible and adaptive systems. Future system shall behave or act appropriately according to situational needs. But this is not guaranteed in novel systems which are complex and their environments change dynamically.

The focus of this paper is to investigate and measure the robustness of coordination mechanisms for multi-agent systems in the context of organic computing. As an application scenario, a traffic intersection without traffic lights is used. Vehicles are modelled as agents.

### *A. Robust Multi-Agent System (RobustMAS)*

An interdisciplinary methodology called “Robust Multi-Agent System” (RobustMAS), has been developed and evaluated regarding different evaluation scenarios and system performance metrics.

The new developed methodology (RobustMAS) has the goal of keeping a multi-agent system robust when disturbances (accidents, unplanned autonomous behaviour) occur. The result is an interaction between decentralised mechanisms (autonomous vehicles) and centralised interventions. This represents a robust hybrid central/self-organising multi-agent system, in which the conflict between a central planning and coordination algorithm on one side and the autonomy of the agents on the other side has to be solved.

The hybrid coordination takes place in three steps:

- A course of action with no disturbance: central planning of the trajectories without deviation of the vehicles.
- Observation of actual trajectories by an Observer component, identifying deviations from plan.
- Replanning and corrective intervention.

In the scenario of this paper, an intersection without traffic lights, the participants are modelled as autonomous (semi-autonomous) agents (Driver Agents) with limited local capabilities. The vehicles are trying as quickly as possible to cross the intersection without traffic lights.

An intersection manager is responsible for coordinating tasks. It performs first a path planning to determine collision-free trajectories for the vehicles (central). This path planning is given to vehicles as a recommendation. In addition, an observation of compliance with these trajectories is done; because the vehicles are autonomous (decentralised) and thus deviations from the plan in principle are possible. Of particular interest is the ability of the system, with minimal central planning intervention, to return back after disturbances to the normal state (robustness).

For the path planning, common path search algorithms are investigated in our earlier paper [1]. Particularly interesting here is the A*- algorithm. The path planning is considered as a resource allocation problem (Resource Allocation Conflict), where several agents move in a shared environment and have to avoid collisions. The implementation was carried out under consideration of virtual obstacles. Virtual obstacles model blocked surfaces, restricted areas (prohibited allocations of resources), which may arise as a result of reservations, accidents or other obstructions. In addition, virtual obstacles can be used for traffic control.

Different types of deviations from the plan of vehicles were examined in our previous paper [1]. The controller is informed by the observer about the detected deviations from the plan, so that it can intervene in time. The controller selects the best corrective action that corresponds to the current situation so that the target performance of the system is maintained.

In this paper, we introduce an appropriate metric for the quantitative determination of the system robustness. The robustness measurement will be made when disturbances (accidents) occur in the system (intersection).

### *B. Measurement of robustness and gain according to the RobustMAS-concept*

Since RobustMAS aims to keep a multi-agent system robust even though disturbances and deviations occur in the system, a new appropriate method to measure the robustness of a multi-agent system is required. The equivalent goal of RobustMAS by the application scenario, a traffic intersection without traffic lights, is to keep the traffic intersection robust even though deviations from the planned trajectories and accidents occur in the intersection. Therefore, a new concept will be introduced in order to define the robustness of multi-agent systems. Additionally, the gain of RobustMAS will be defined and used to show the benefit of the system that can be obtained through using the hybrid central/self-organising concept, which is a hybrid coordination (central and decentral), compared to using only a decentral planning.

According to the RobustMAS-concept, the robustness of a multi-agent system can be defined as follows:

“The robustness of a multi-agent system is the degradation of the system performance under disturbances that take place in the system environment and under deviations from the plan (central) that occur in the behaviour of the agents (autonomous, decentral)”.

Consequently, RobustMAS-concept assumes that a robust system keeps its performance acceptable after occurrence of disturbances and deviations from the plan.

In order to measure the robustness of RobustMAS in the traffic intersection system, the throughput metric is used for determining the reduction of the performance (system throughput) of RobustMAS after disturbances (accidents) and deviations from the planned trajectories occur. That is because throughput is one of the most commonly used performance metrics. Therefore, the comparison of the throughput values is required in the three cases: without disturbance, with disturbance with intervention and with disturbance without intervention.

Figure 2 illustrates this comparison where ($t_1$) is the simulation time step at which the disturbance (accident) will occur and remain until the simulation time step ($t_2$). This figure shows cumulative performance (throughput) values of the system before and after disturbance occurrence comparing the three mentioned cases.

The black curve is the performance (throughput) of the system if no disturbance occurs during the simulation. The green curve is the performance of the system when a disturbance at time ($t_1$) occurs and the central planning intervenes on time. The simulation lasts until time ($t_2$). The red curve is the performance of the system when a disturbance at time ($t_1$) occurs and the central planning does not intervene. Here, two areas can be distinguished: Area1 and Area2 in order to measure the robustness of RobustMAS as depicted in Figure 3.

This figure shows the idea of how the robustness of the system as well as the gain of the system can be determined according to the RobustMAS-concept.

The robustness (R) of a system (S) can be determined as described in the next formula:

$$Rs = \frac{\text{Area2}}{\text{Area1} + \text{Area2}} = \frac{\int_{t1}^{t2} Per(t)_{(withIntervention)}.d(t)}{\int_{t1}^{t2} Per(t)_{(NoDisturbance)}.d(t)}$$

This means that the robustness is the surface area 2 divided by the sum of the two surface areas 1 and 2. The Area2 is the integral of the green curve (disturbance with intervention) between t= $t_1$ and t= $t_2$. The sum of Area1 and Area2 is the integral of the black curve (no disturbance) between t= $t_1$ and t= $t_2$.

Additionally, the gain of the system can be used as a secondary measure. In this context, the gain of a system can be defined according to the RobustMAS-concept as follows:

“The gain of a system is the benefit of the system through central planning (compared to decentral planning). Accordingly, the gain of a system represents the difference between the system performance (throughput) in the two cases, with and without intervention of the central planning algorithm”. This issue is expressed by the following equation:

$$\text{Gain} = \Delta\text{Per (Intervention)} - \Delta\text{Per (NoIntervention)}$$

As depicted in Figure 3, the gain of the system can be calculated using the values of the system performance (throughput values) at the end of the simulation time ($t=t_2$). Here, ΔPer(Intervention) represents the difference between the system performance in the two cases, without disturbance and disturbance with intervention of the central planning algorithm; whereas ΔPer(NoIntervention) represents the difference between the system performance in the two cases, disturbance with and without intervention of the central planning algorithm.

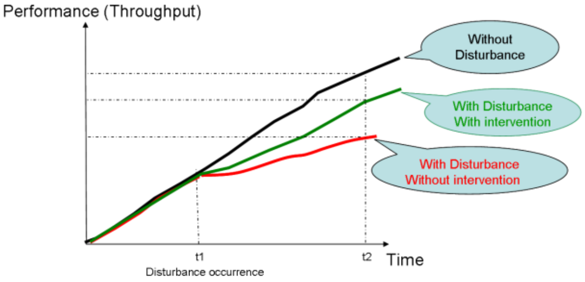

Figure 2. The comparison of system performance (throughput) by the situation (without disturbance) to the situation disturbance in both cases (with and without intervention)

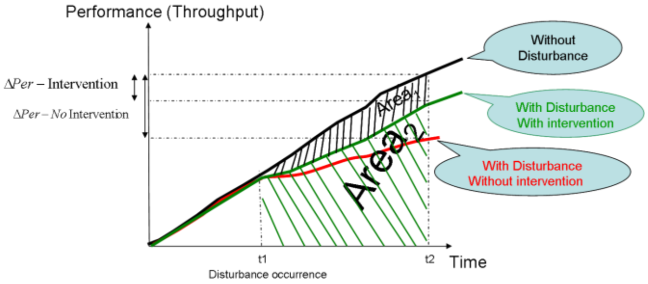

Figure 3. Measuring robustness and gain according to the RobustMAS-concept

The discussion of the robustness measurement using the system throughput metric will be based on the simulation parameter, the disturbance strength. This parameter represents the size of the accident in the used traffic system.

Obviously the disturbance strength influences the system performance which in turn leads to different degrees of system robustness. When the disturbance strength is increased, then the system robustness will reduce. This means that the increase of the disturbance strength is inversely proportional to the degree of the system robustness.

According to the used application scenario, the size of the accident influences the intersection throughput (the number of vehicles that have left the intersection area) which in turn leads to different degrees of the robustness of the intersection. When the size of the accident increases, then the intersection robustness will decrease. This can be justified simply on the ground that accidents will cause obstacles for the vehicles in the intersection. These obstacles will impede the movement of vehicles which are behind the accident location. Additionally, the central plan algorithm considers the accidents as virtual obstacles (restricted areas) and therefore it limits the planned trajectories of potential traffic. The autonomous vehicles which do not obey their planned trajectories have to avoid the accident location by performing a lane change (to the right or to the left of the accident location) if it is possible. Certainly, autonomous vehicles have to check the possibility to avoid the accident by pulling into another lane before they take this evasive action. So, the vehicle behind the accident location tries to overtake the accident location on the right if the intended position is not occupied by another vehicle. Otherwise, if the intended position is occupied by another vehicle, then the vehicle tries to overtake the accident location on the left if the intended position is not occupied by another vehicle. If all potential intended positions are occupied, then the vehicle stops (doesn’t change its position) and repeats this behaviour (the evasive action) again in the next simulation step.

## V. PERFORMANCE EVALUATION

In this section, we present a complete empirical evaluation of our system using the model of a traffic intersection, which was designed and described in our earlier paper [1]. This evaluation includes experiments for measuring the robustness of the system, in which deviations from plan occur and disturbances (accidents) appear in the intersection system. That means, it deals with deviations from planned (desired) behaviour of agents (vehicles), in addition to disturbances (accidents).

### *A. Test situation*

In this test situation, the vehicles do not obey their planned trajectories (the central plan) and thus deviations from the plan will occur as well as accidents in the intersection.

In this regard, an observation of actual trajectories by the observer will be made in order to detect any deviations from plan and to detect potential accidents in the intersection allowing the controller to make replanning for all affected trajectories using the path planning algorithm. This will be carried out via the deviation detector component and the accident detector component in the observer [1][2].

The test situation serves to measure the robustness of the intersection system and to assess the degree of the robustness of RobustMAS during disturbances (e.g., accidents) and deviations (e.g., unplanned autonomous behaviour).

### *B. Measuring robustness and gain*

As mentioned above, the throughput metric is used to determine the reduction of the performance (system throughput) of RobustMAS after disturbances (accidents) occur and consequently to measure the robustness of RobustMAS in the traffic intersection system. Additionally, the discussion of the robustness measurement is based on the simulation parameter, the disturbance strength (the size of the accident). The measurement has been repeated in the cases that the disturbance strength is 1, 2, and 4. That means, the accident occupies an area of size 1, 2 and 4 cells in the traffic intersection. The results were obtained in an interval between 0 und 3000 ticks.

It can be concluded that the increase in the size of the accident is inversely proportional to the degree of the intersection robustness.

RobustMAS tries to guarantee a relatively acceptable reduction of the intersection robustness when the size of the accident increases. RobustMAS ensures at least that increasing of size of the accident will not lead to failure of the intersection.

Because the location of the accident within the intersection plays a major role in the performance of the intersection system, the simulation was repeated 10 times. Each time of repetition, an accident will be generated in a random position of the intersection by choosing a random (x, y) coordinate pair within the intersection. This (x, y) coordinate pair represents the central cell of the accident. The other cells which represent the whole accident location will be chosen also randomly depending on the value of the simulation parameter “size of accident”, so that the chosen cells will surround the central cell (x, y) of the accident. So, it can be ensured that accidents will be generated in different parts of the intersection achieving more realistic study. The average values of the system throughput will be calculated from several repetitions of the simulation (random accident locations), so that a picture of how an accident would affect the system performance is created.

The simulation parameter “Disturbance occurrence time” (Accident occurrence time) represents the time (the time step in the simulation) at which the accident will be generated. The time is measured by ticks. In the simulation, the “Accident tick” was adjusted to the value of the tick “1000”, i.e., an accident should be generated at tick “1000”. That means, the simulation has no accident in the interval [0-1000]; whereas it has an accident in the remaining simulation interval [1000-3000] as depicted in Figure 4. Here, the system performance is the intersection throughput. The throughput is measured by the number of vehicles that left the intersection area (cumulative throughput values).

The robustness and the gain of the traffic intersection system can be determined using the two formulas of the robustness and the gain of the system described above.

In order to see the effect of the disturbance strength (size of the accident), Table I compares the obtained results of the robustness and the gain of the system for various values of disturbance strength after 3000 ticks.

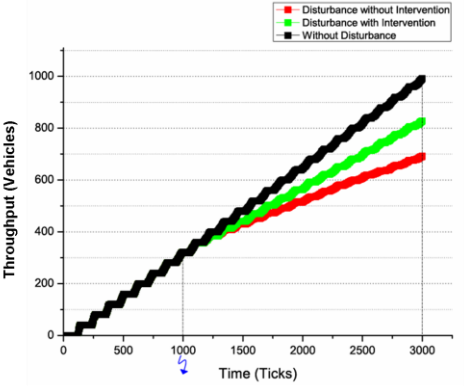

Figure 4. The “Disturbance occurrence time” adjusted to the tick (1000) and the simulation length is (3000) ticks

| **Disturbance strength (Accident size)** | **Robustness (%)** | **Gain (Vehicles)** |
|---|---|---|
| 1 | 87 | 137 |
| 2 | 86 | 161 |
| 4 | 83 | 169 |

TABLE I. THE ROBUSTNESS AND THE GAIN OF THE SYSTEM FOR VARIOUS VALUES OF DISTURBANCE STRENGTH

It can be concluded that when the disturbance strength increases, the robustness of the system decreases, but very slightly showing a high degree of robustness. This emphasises that a degradation of the system throughput was established when an accident was occurred in the intersection and the vehicles made deviations violating their planned trajectories. Therefore, in case of disturbances (accidents), the intervention of the central plan (a central planning algorithm) led to better system performance than the decentralised solution in which agents (vehicles) have to plan locally their trajectory.

On the other hand, when the disturbance strength increases, the gain of the system increases. This confirms the conclusion that the intervention of the central plan was better demonstrating an improvement of the system throughput.

Therefore, it is inferred that a global problem (e.g., an accident in the intersection) should be solved at global level, because there is a central unit (the o/c architecture) that has the global view of the system. This central unit can plan better than a decentral unit. A central unit needs only longer time than a decentral unit. This issue can be solved simply by providing central units that have plenty of resources, e.g., CPU capacity (real-time requirements), memory capacity, etc, as well as the management of these resources.

## VI. CONCLUSIONS

In this paper, we extended the implementation of the generic o/c architecture adapted to our traffic scenario and accomplished our experiments assuming that accidents (disturbances), in addition to deviations from plan, occur in the system environment (intersection).

Additionally, we introduced an interdisciplinary methodology called “Robust Multi-Agent System” (RobustMAS). We developed and evaluated RobustMAS aiming to keep a multi-agent system robust when disturbances (accidents, unplanned autonomous behaviour) occur. RobustMAS represents a robust hybrid central/self-organising multi-agent system, in which the conflict between centralised interventions (central planning) and the autonomy of the agents (decentralised mechanisms, autonomous vehicles) was solved.

In this regard, we measured the system performance and compared the two cases, the system performance with disturbances on one side and the system performance without disturbances from the other side. This comparison showed that the system performance remains effective (robust) despite disturbances and deviations occurred in the system. Furthermore, we presented an appropriate metric for the quantitative determination of the robustness of such hybrid multi-agent systems. Subsequently, we measured the robustness and gain of a multi-agent system using the RobustMAS-concept. The experiments showed a high degree of robustness of RobustMAS.

## VII. FUTURE WORK

One aspect that may be of interest for future work is the fairness between the system’s agents (vehicles). In order to achieve this fairness, there are different approaches that deal with this issue. The other aspect that will be an important issue in future is the coordination and cooperation of multiple intersections without traffic lights.

## REFERENCES

[1] Yaser Chaaban, Jörg Hähner, and Christian Müller-Schloer. “Towards fault-tolerant robust self-organizing multi-agent systems in intersections without traffic lights”. In Cognitive09: proceedings of The First International Conference on Advanced Cognitive Technologies and Applications, November, 2009, pp. 467-475, Greece. IEEE.

[2] Yaser Chaaban, Jörg Hähner, and Christian Müller-Schloer. “Towards Robust Hybrid Central/Self-organizing Multi-agent Systems”. In ICAART2010: proceedings of the Second International Conference on Agents and Artificial Intelligence, Volume 2 , pp. 341-346, January 2010 Spain.

[3] CAS-wiki: Organic Computing. http://wiki.cas-group.net/index.php?title=Organic_Computing, [retrieved: February, 2012].

[4] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots, pp. 4:226–234, 1997.

[5] Christian Müller-Schloer. “Organic computing: on the feasibility of controlled emergence”. In CODES+ISSS ’04: Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, pp. 2–5. ACM, 2004.

[6] Urban Richter, Moez Mnif, Jürgen Branke, Christian Müller-Schloer, and Hartmut Schmeck. “Towards a generic observer/controller architecture for organic computing”. In Christian Hochberger and Rüdiger Liskowsky, editors, INFORMATIK 2006 - Informatik für Menschen!, volume P-93 of GI-Edition - Lecture Notes in Informatics (LNI), pp. 112–119. Bonner Köllen Verlag, 2006.

[7] Kurt Dresner and Peter Stone. “Mitigating catastrophic failure at intersections of autonomous vehicles”. In AAMAS ’08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems, pp. 1393–1396, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[8] H. Schmeck, C. Müller-Schloer, E. Cakar, M. Mnif, U. Richter. “Adaptivity and Self-organisation in Organic Computing Systems”. ACM Transactions on Autonomous and Adaptive Systems, 2009, pp. 10:1-10:32.

[9] V. Shestak, H. J. Siegel, A. A. Maciejewski, and S. Ali. “The robustness of resource allocations in parallel and distributed computing systems”. In Proceedings of the International Conference on Architecture of Computing Systems (ARCS 2006), pp. 17–30, 2006.

[10] D. England, J. Weissman, and J. Sadagopan. “A new metric for robustness with application to job scheduling”. In IEEE International Symposium on High Performance Distributed Computing 2005 (HPDC-14), Research Triangle Park, NC, July 24-27, 2005.

[11] Waldschmidt, K., Damm, M., “Robustness in SOC Design”, Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on Digital System Design, pp. 27-36, Volume: Issue: , 0-0.