RobustMAS: 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
e-mails: {chaaban, cms}@sra.uni-hannover.de

Jörg Hähner
Institute of Organic Computing
University of Augsburg
Augsburg, Germany
e-mail: 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 at a desired performance level when deviations from planned (desired) behaviour occur in the system (robustness). 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 analyse two previous quantitative approaches presented, among others, in the literature towards a generalised robustness metric. Furthermore, 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. The experimental results demonstrated a high degree of the robustness of the developed concept against disturbances.**

Keywords-Robustness; Organic Computing; Hybrid Coordination; Multi-Agent Systems; Performance measurement systems

I. INTRODUCTION

This article is an extension of a previously published paper [1]. 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 for technical systems aiming to cope with the increasing complexity [2][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.

Organic systems or autonomic systems [4][5] try to realise quality in several aspects of system engineering including: functional correctness, safety, security, robustness/reliability, credibility, and usability [6][7].

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 development 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. A 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, a 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) [8]. 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 elements 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 traits simultaneously. This means that a conflict between a central controller (e.g., a coordination algorithm) and the autonomy of the system components must be solved in order to achieve 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 [9], the behaviour of OC systems can be observed and controlled. A generic O/C architecture was presented in [10] 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 Systems (ITS)" was developed. ITS have several applications in traffic and automotive engineering. According to ITS, numerous notions were distinguished such as, among others, intelligent vehicles, intelligent intersections, and autonomous vehicles. In this context, a traffic intersection without traffic lights was chosen 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. However, the basic idea of a hybrid approach is applicable for other systems as well.

This paper is organised as follows. Section II describes our original system introduced in [11][12]. Section III 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 IV 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 V introduces the evaluation of the system performance by means of experimental results. Section VI draws the conclusion of this work. Finally, the future work is explicated in Section VII.

II. The Original System

This paper is an extended version of our conference paper [1] presented at Cognitive2012. With respect to [1], this paper presents an expanded discussion of related work, allowing us to analyse two previous quantitative approaches towards a generalised robustness metric. Furthermore, the robustness measurement will be considered in two ways in this paper, while there was only one way in [1]. Finally, this paper shows detailed version of results using cumulative throughput values in upper figures and throughput values per time unit in lower figures.

In previous papers, we introduced a system for coordinating vehicles at a traffic intersection using an O/C architecture [11][12]. 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 [11] [12].

Moreover, specifying the desired behaviour of agents in a shared environment was considered in [13]. So, we presented a convenient method to achieve such desired behaviour. For this purpose, A*-algorithm for path planning of agents (vehicles) was proposed [13].

Additionally, handling of deviations from planned (desired) behaviour was studied in [14]. To address this issue, we extended our prototype implementation with the aim of making it capable of handling deviations from planned behaviour. In this way, the hybrid central/self-organising concept tolerates that some agents behave autonomously. Here, the autonomy of the agents is recognised as a deviation from the plan of the central algorithm, if the agents are not respecting this plan [14].

Furthermore, we provided an overview of a several robustness approaches in multi-agent systems (MAS) in [15]. The survey is concerned with MAS in a variety of research fields.

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 completely.

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

III. State of the Art

Keeping a system at a desired performance level in presence of disturbances or deviations from plan has been 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.

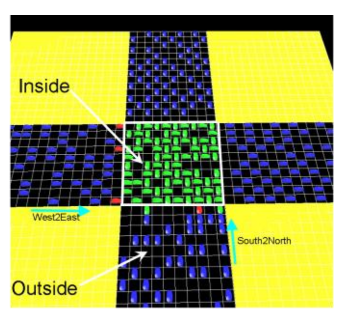

Figure 1. The traffic intersection without traffic lights

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 [16]. 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*

In order to have the ability to design robust multi-agent systems, robustness metrics are required. These metrics play the role to mitigate the expected degradation of the system performance when any disturbances occur. Many research projects deal with system robustness. Their objective is to measure robustness and to find an appropriate metric for it. These projects are in various kinds of science.

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 proposed. Every robustness measure 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 the definition of a 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 a system as well as the maximum deviation from a benchmark where the system has still the ability to deal with failures [17].

B. *Generalised robustness metric*

Viable quantitative approaches in order to measure robustness are required. Some approaches were introduced, among others, in [18][19][20]. Among those, both the FePIA (Features Perturbation Impact Analysis) procedure in [18] and the statistical approach in [19] 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 the environment of the system. Additionally, these metrics were mathematically described. Both approaches in [18] and in [19] are applicable in embedded systems design [20] where embedded systems are designed as Systems on Chip (SoC).

In the following, the FePIA procedure and the statistical approach will be explained.

1) *FePIA procedure*

The FePIA procedure is presented in [18] in order to derive a robustness metric so that it can be used for an arbitrary system. The authors there discussed the robustness of resource allocations in parallel and distributed computing systems. Consequently, a derived metric from the FePIA procedure was designed for a certain allocation of independent applications in a heterogeneous distributed system demonstrating the utility of the robustness metric. Here, the goal was to maximise the robustness of the produced resource allocations. Moreover, the authors have defined the robustness (indeed, a resource allocation is to be robust) as a restricted degradation of the system performance against uncertainties (perturbations) in specified system parameters.

FePIA stands for Features Perturbation Impact Analysis. The FePIA procedure defines a schema that presents a robustness-radius for the system based on a tolerance region. This procedure identifies four general steps [18][20]:

1. The important system performance features f_i that may cause degradation of the system performance. They are combined into a feature vector Φ: $\Phi = \{\varphi_1, \ldots, \varphi_n\}$.
2. The perturbation parameters: $\pi = \{\pi_1, \ldots, \pi_m\}$.
3. The impact of perturbation parameters on system performance features. This is modelled with individual functions $f_{ij} : \pi_i \rightarrow \varphi_j$, selecting a tolerance region (β_j^{min}, β_j^{max}) for each φ_j (see Figure 2).
4. The analysis (it analyses the values of π_i) to determine the degree of robustness.

The main point here is to produce a mathematical relationship between the system performance features and the perturbation parameters (in the sense of the impact). After that, a variation in the perturbation parameters, which lead to a performance degradation exceeding the allowable performance limits (tolerance region), can be detected. This variation represents the robustness radius (optimisation problem) [19].

So, r (φ_j , π_i) represents the robustness-radius of the system according to the system performance feature φ_j and the perturbation parameter π_i. Accordingly, in order to calculate the robustness of the whole system in the case of a certain perturbation parameter, the minimum across all features of system performance has to be found. Figure 2 illustrates the FePIA procedure.

Here, a tolerance region is defined by a lower boundary ($\beta^{\min}$) and an upper boundary ($\beta^{\max}$), which can be expressed as in the next formulas:

$$\beta^{\min} = \min \ \left\{ \ f\left(\pi^{orig} - r\right),\ f\left(\pi^{orig} + r\right) \ \right\} \qquad (1)$$

$$\beta^{\max} = \max \ \left\{ \ f\left(\pi^{orig} - r\right),\ f\left(\pi^{orig} + r\right) \ \right\} \qquad (2)$$

A robustness definition for analog and mixed signal systems was derived in [20] using the FePIA procedure. The author has evaluated the proposed robustness formula applying affine arithmetic (modelling the deviations by affine

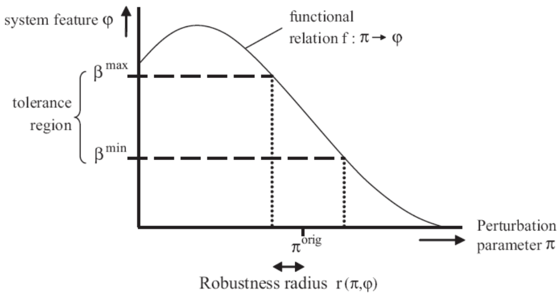

Figure 2. The general FePIA procedure [20]

expressions as in [21]) with a semi-symbolic simulation. The symbolic representation used in semi-symbolic simulations makes designers aware of the contribution of uncertainty to the deviation at the output of the simulated system. Also, the outcomes of the simulation are affine expressions, which semi-symbolically represent possible deviations [21].

As a result, a robustness definition for analog and mixed signal systems was derived that is based on the estimation of precision versus the robustness radius using the FePIA procedure as described in the next formula:

$$robustness(\varphi,\pi) := \frac{r(\varphi,\pi)}{rad(\pi)} \quad (3)$$

where $rad(\pi)$ characterises the confidence interval of deviations from π [20].

According to this formula, which can be used in the design phase, three cases can be considered.

- First, the robustness is less than 1 and hence the system is not robust and it may fail.
- Second, the robustness is equal to 1 and therefore the system is robust to some extent and it fulfils the minimum requirements.
- Third, the robustness is greater than 1 and hence the system is robust against additional deviations [20].

The drawback of the FePIA procedure is that the tolerance regions (the limits of the performance features) are arbitrarily selected. Thus, the FePIA procedure is applicable for systems where the system performance and the tolerable deviations can be well-defined [20].

2) Statistical approach

The statistical approach has been introduced by England et al. in [19] to obtain a type of robustness metric, which can be used for an arbitrary system. The authors there present a methodology aiming to characterise and measure the robustness of a system (using a quantitative metric) in the face of a specific disturbance (perturbation).

The authors define robustness as follows: “Robustness is the persistence of certain specified system features despite the presence of perturbations in the system’s environment.” [19].

Similar to the FePIA procedure, system performance features in the statistical approach will be taken into consideration versus the perturbation size (disturbance size). Therefore, the intention of the authors was to measure the amount of degradation of the system performance relative to the perturbation size [20][19]. For this purpose, the cumulative distribution function (CDF) of a system performance feature is used. CDF is the proportion of observations less than or equal to a specified value (x) when a set of performance observations (X) is given [19]. The robustness can be determined according to the difference between functions F and F*. The function F is the CDF of a performance feature in the case of normal operating conditions; whereas the function F* is the CDF of a performance feature in the case of perturbations.

The maximum distance between F and F* represents the amount of performance degradation. This distance (δ) was computed by means of the Kolmogorov-Smirnov (K-S) statistic (sup is the supremum):

$$\delta = \sup_{-\infty<x<\infty} \left(F(x) - F^*(x)\right) \quad (4)$$

Moreover, the distance (δ) has to be weighted with a weighting function (to compensate for the underestimation of δ) producing the adjusted K-S statistic (δ_w):

$$\delta_w = \sup_{-\infty<x<\infty} \left(F(x) - F^*(x)\right)\Psi(x) \quad (5)$$

The advantage of this method is that it considers the complete distribution of system performance (performance observations); whereas other methods consider only average measurements. In this context, it can be inferred that the system is robust against the applied perturbation when the distance between F and F* (the amount of performance degradation) is very small. Therefore, the smaller the distance is, the more robust the system becomes. Figure 3 illustrates the statistical approach (the adjusted K-S statistic) [19].

In Figure 3, the robustness of a system is characterised by the measurement of δ_w as a function of the applied perturbation size (in other words, by the gradient of δ_w relative to the amount of perturbation experienced [20]). This means that this system can withstand different levels of perturbation. Here, three cases can be recognised.

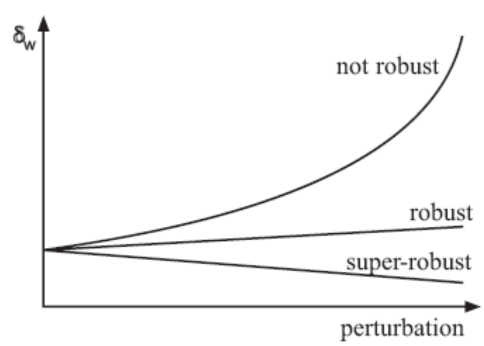

Figure 3. Characterising the robustness of a system according to the statistical approach [20]

First, the robust system, wherein δ_w exhibits a slight increase with increasing the perturbation size. Second, non-robust system, wherein δ_w shows a great (probably non-linear) increase with increasing the perturbation size. Third, the super-robust system, wherein δ_w exhibits a slight decrease with increasing the perturbation size. The perturbation in the last case is a profitable perturbation (see [19] for an example).

According to [20], the proposed robustness metric based on the statistical approach is appropriate to use in the design process, where it acts as absolute robustness indicator for profiling targets. In this case, specifications must be executable, so that simulations can be carried out to supply an adequate amount of statistical data.

Comparing with the FePIA procedure, this methodology is generally applicable to various classes of computing systems. Also, it is easier to determine the robustness. That means, the statistical approach has avoided the drawback of the FePIA procedure, so that a tolerance region needs not to be formed. Additionally, they employed their methodology in three applications of job scheduling: backfilling jobs on supercomputers (parallel machines), overload control in a streaming video server, and routing requests in a distributed network service. The third application shows the role of robustness to obtain improvements in system design. Additionally, as mentioned above, this robustness metric would have the advantage of the consideration of the complete distribution of system performance.

C. Summary: Measures for robustness

Several research projects propose diverse measures of system robustness. These projects measure robustness according to their definition of the robustness in different application areas. In this context, some quantitative approaches were used, such as the FePIA procedure in [18] and the statistical approach in [19]. However, there is a clear lack of study of the robustness metrics in designing robust multi-agent systems in technical systems. Therefore, there still is the question how the robustness can be guaranteed and measured in technical systems. As a result, both approaches discussed above do not comply with the RobustMAS concept introduced in this paper to characterise robustness.

This non-compliance can be traced back to the fact that RobustMAS focuses on the robustness of hybrid central/self-organising multi-agent systems. For this purpose, RobustMAS proposes the concept of relative robustness for measuring the ability to maintain a specific minimum level of system performance (a desired performance level) in the presence of deviations from desired behaviour (e.g., unplanned autonomous behaviour) and disturbances in the system environment. Based on this, according to the RobustMAS concept, robustness is the ability of the system, with minimal central planning intervention, to return after disturbances (internal and external changes) to the normal state.

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 technical systems. Future systems shall behave appropriately according to situational needs. But this is not guaranteed in novel systems, which are complex and act in dynamically changing environments.

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 running at a desired performance level when disturbances (accidents, unplanned autonomous behaviour) occur (for details see Definition 4: *Disturbance strength*). 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 hand and the autonomy of the agents on the other has to be solved.

The hybrid coordination takes place in three steps:

1. A course of action with no disturbance: central planning of the trajectories without deviation of the vehicles.
2. Observation of actual trajectories by an Observer component, identifying deviations from plan.
3. 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, since 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 after disturbances to the normal state.

For the path planning, common path search algorithms are investigated in our earlier paper [11]. 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.

In [13], we focused on planning of the desired behaviour of agents in a shared environment. Based on this, an adapted A*-algorithm for path planning of agents has been applied. The adaptation was necessary for the requirements of the used traffic scenario, because a vehicle can only take a "rational" path, whereas an agent (e.g., robot) can take any calculated path. Consequently, the designed algorithm calculates collision-free trajectories (central planning) for all agents (vehicles) in a shared environment (the centre of the intersection) enabling them to avoid collisions. The experimental results demonstrated a high performance of our adapted A*- algorithm.

Different types of deviations of the vehicles from the plan have been investigated in our previous paper [11]. 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 at a desired performance level 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 at a desired performance level 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 hybrid central/self-organising concept.

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

Definition 1: Robustness.

"A (multi-agent) system is considered robust against disturbances if its performance degradation is kept at a minimum".

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

Definition 2: Relative robustness.

"The relative robustness of a (multi-agent) system in the presence of a disturbance is the ratio of the performance degradation due to the disturbance divided by the undisturbed performance".

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. 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:

(1) Without disturbance.
(2) With disturbance with intervention.
(3) With disturbance without intervention.

Based on this, the robustness measurement of RobustMAS will be considered in two ways:

- Using cumulative system performance, i.e., cumulative throughput (# Agents), where the system is considered only until the time when the disturbance ends.
- Using system performance, i.e., throughput per time unit (# Agents/sec), where the system is considered until the time when the system returns after disturbances to its normal state like before.

For this explanation of the robustness measurement, the words agent and vehicle can be used interchangeably.

1) Using cumulative system performance (cumulative throughput)

Figure 4 illustrates this comparison where t_1 is the time at which the disturbance (accident) occurs. The disturbance is assumed to remain active until the time t_2. This figure shows the cumulative performance (throughput) values of the system before and after the disturbance comparing the three mentioned cases.

The black curve is the performance (throughput) of the system if no disturbance occurs. 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 system is considered until time t_2 when the disturbance ends. 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: $Area_1$ and $Area_2$ in order to measure the robustness of RobustMAS as depicted in Figure 5.

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.

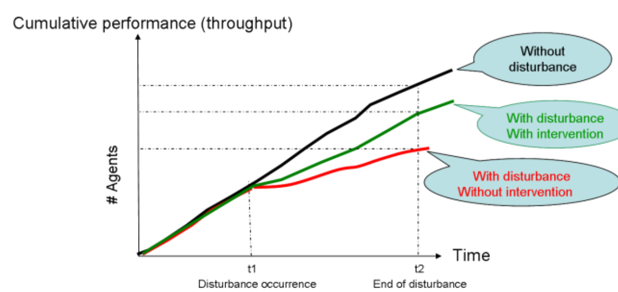

Figure 4. Comparison of cumulative system performance (throughput) for three situations

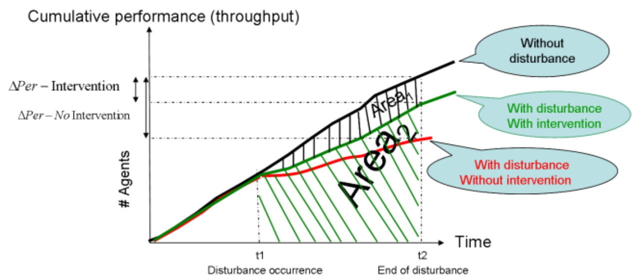

Figure 5. Measuring robustness and gain using cumulative system performance

The relative robustness (R) of a system (S) is determined as follows:

$$R = \frac{\text{Area}_2}{\text{Area}_1 + \text{Area}_2} = \frac{\int_{t1}^{t2} Per(t)_{(withIntervention)} d(t)}{\int_{t1}^{t2} Per(t)_{(NoDisturbance)} d(t)} \tag{6}$$

This means that the robustness is Area_2 divided by the sum of the two areas 1 and 2. Area_2 is the integral of the green curve (disturbance with intervention) between t_1 and t_2. The sum of Area_1 and Area_2 is the integral of the black curve (no disturbance) between t_1 and 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:

Definition 3: Gain.

"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:

$$Gain = \Delta Per_{(NoIntervention)} - \Delta Per_{(Intervention)} \tag{7}$$

As depicted in Figure 5, the gain of the system can be calculated using the values of the system performance (throughput values) at the time t_2. Here, $\Delta Per(Intervention)$ represents the difference between the system performance in the two cases, without disturbance and disturbance with intervention of the central planning algorithm; whereas $\Delta Per(NoIntervention)$ represents the difference between the system performance in the two cases, disturbance with and without intervention of the central planning algorithm.

2) Using system performance (throughput per time unit)

In this case, the system performance, i.e., throughput per time unit (# Agents/sec) is used. Additionally, the system is considered longer than in the case of the cumulative performance (cumulative throughput) values. Therefore, compared to that case that defines time t_1, the occurrence time of disturbance, and time t_2, the end time of disturbance,

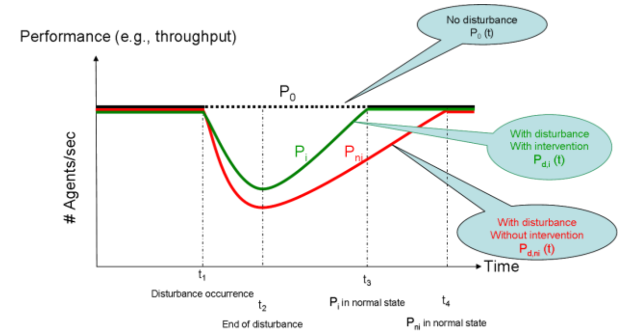

Figure 6. Comparison of system performance (throughput per time unit) for three situations

the times t_3 and t_4 will also be defined. Here, t_3 is the time at which the system returns to its normal state with minimal central planning intervention, while t_4 is the time at which the system returns to its normal state without central planning intervention. In this regard, the normal state represents the system performance level at its best when no disturbances occur (under normal operating conditions).

Here, we use the following functions:

- P_0 (t): represents the system performance when no disturbances occur (normal state).
- $P_{d,\,ni}$ (t): represents the system performance with a disturbance with no intervention by the central planning.
- $P_{d,\,i}$ (t): represents the system performance with a disturbance with an intervention of the central planning.

Figure 6 shows the performance (throughput per time unit) values of the system before and after the disturbance until the time when the system returns to its normal state like before comparing the three mentioned cases.

In accordance with the definition 2 mentioned above, the relative robustness (R) of a system (S) is determined as follows:

$$R = \frac{\int_{t1}^{t4} P_{d,i}(t) d(t)}{\int_{t1}^{t4} P_0(t) d(t)} \; ; \quad 0 \le R \le 1 \tag{8}$$

Here, the lower and upper boundaries can be set as follows:

- R = 0 represents the lower boundary case of the relative robustness, where the system is considered as non-robust against disturbances (very poor performance). It appears when $P_{d,\,i}$ (t) << P_0 (t), i.e., the performance degradation is very strong due to the disturbance in spite of the intervention, compared to the performance when no disturbance occurs. Thus, the system behaviour is not acceptable in the face of disturbances.
- R = 1 represents the upper boundary case of the relative robustness, where the system is considered as strongly robust against disturbances (an optimal performance, an ideal behaviour). It occurs, when

$P_{d,i}$ (t) = P_0 (t), i.e., there is no performance degradation due to the intervention despite the presence of disturbances.

Furthermore, the system could be also weakly robust if its performance level is acceptable but not optimal in the presence of disturbances. Therefore, the system behaviour is acceptable but not ideal.

Similar to the definition 3 mentioned above, the gain of a system is determined as the difference between the performance in both cases, disturbances with and without intervention:

$$\begin{aligned} Gain(i \rightarrow ni) &= \#Agents(i) - \#Agents(ni) \\ &= \int_{t1}^{t4} [P_{d,i}(t) - P_{d,ni}(t)]\, d(t) \end{aligned} \tag{9}$$

Consequently, the loss of a system is determined as the difference between the performance in both cases, no disturbance and disturbances with intervention:

$$Loss = \int_{t1}^{t4} [P_0(t) - P_{d,i}(t)]\, d(t) \tag{10}$$

The discussion of the robustness measurement using the system throughput metric will be based on the parameter *disturbance strength*. In this regard, the disturbance strength can be defined according to the RobustMAS concept as follows:

Definition 4: Disturbance strength.

"A disturbance strength is a positive constant defining the strength (size) of the disturbance".

This parameter represents the size of the accident in the used traffic system. Accordingly, the robustness measurement was 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 as depicted in Figure 7.

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 performance will be reduced. This means that the increase of the disturbance strength is inversely proportional to the degree of the system robustness.

However, the definition of system robustness can be extended to include the strength of disturbances experienced (amount of disturbances applied). Accordingly, the robustness (Rob) of a given system depending on the disturbance strength ($Dist_{str}$) can be determined in formula (11).

This means that Rob = R $*$ $Dist_{str}$, where R is the relative robustness defined above. In this case, the integral will be between the time t_1 at which the disturbance begins, and time t_2, at which the disturbance ends. This formula implies that a system shows varying degrees of robustness (Rob) while the disturbance strength is varied.

$$Rob = \underbrace{\frac{\int_{Start\,dist.}^{End\,dist.} P_{d,i}(t)\,d(t)}{\int_{Start\,dist.}^{End\,dist.} P_0(t)\,d(t)}}_{R} * Dist_{str} \tag{11}$$

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 performance 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 as depicted in Figure 8. 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 (does not change its position) and repeats this behaviour (the evasive action) again in the next simulation step.

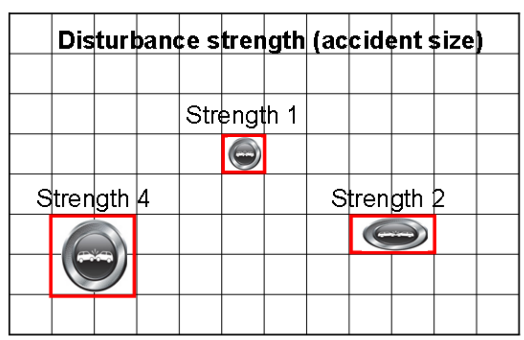

Figure 7. The disturbance strength (the accident size) in three cases: 1, 2, and 4 cells in the traffic intersection

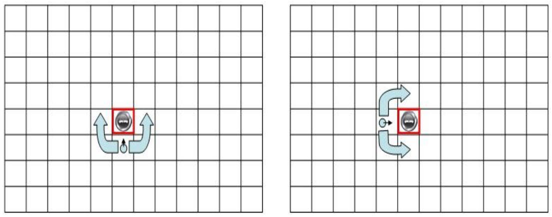

Figure 8. The evasive action of autonomous vehicles that check the possibility (right or left) to avoid the accident by pulling into another lane

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 [11]. 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 [11][12].

The test situation serves to measure the robustness of the traffic 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) and consequently to measure the robustness of RobustMAS in the intersection system. Additionally, how the discussion of the robustness measurement is carried out depends on the disturbance strength $Dist_{str}$ (the size of the accident) involved in the experiments. As illustrated in Figure 7, $Dist_{str}$ is varied (1, 2 or 4). The results were obtained in an interval between 0 und 3000 ticks, where the maximum number of vehicles (V_{max}) is 40 vehicles in both directions and the traffic level (TL) is 5 vehicles/tick in each direction.

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 in 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 9. 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 in the upper figure or throughput values per time unit in the lower figure).

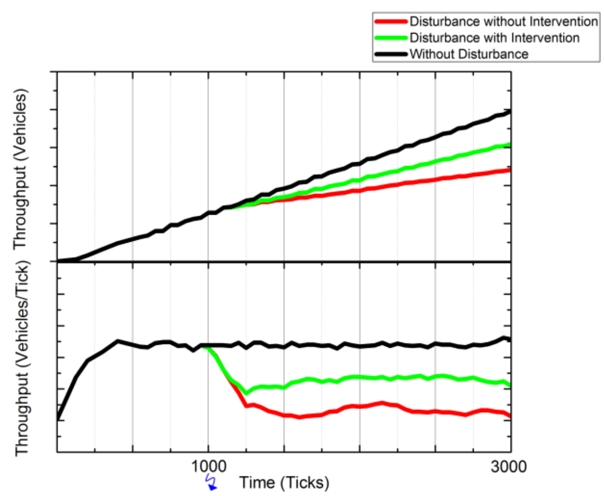

Figure 9. The “Disturbance occurrence time” adjusted to the tick 1000 and the simulation length is 3000 ticks (upper figure is cumulative throughput; lower figure is throughput per time unit)

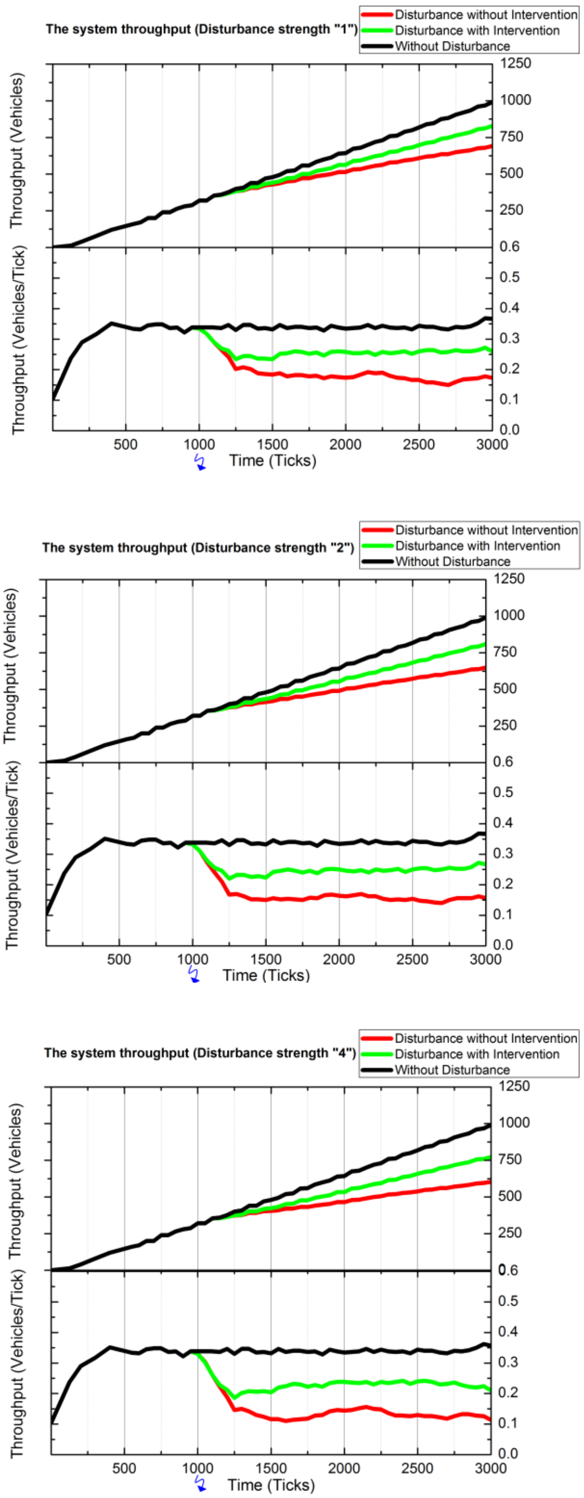

Figure 10. The system throughput per time unit (lower figure) and the cumulative system throughput (upper figure) using different values of the disturbance strength (size of the accident)

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

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

The upper figure of Figure 10 shows the cumulative system performance values (throughput) of the intersection system in an interval between 0 und 3000 ticks comparing the three mentioned cases (without disturbance, disturbance without intervention and disturbance with intervention) using various values of the disturbance strength (size of the accident). Furthermore, the lower figure of Figure 10 shows the same as the upper figure using the throughput per time unit (# Vehicles/tick).

The robustness and the gain of the traffic intersection system can be determined using the two formulas of the relative robustness (R) 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.

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 has 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 sufficient resources, e.g., CPU capacity (real-time requirements), memory capacity, etc, as well as the management of these resources.

VI. CONCLUSION

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 at a desired performance level 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 discussed two quantitative approaches introduced in the literature to quantify robustness. Afterwards, 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 the 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. Finally, since the RobustMAS concept is applicable for other systems, this paper leaves space for the applicability of the RobustMAS concept for shared spaces. The current traffic scenario used in this work has similarities to shared spaces in the working environments and conditions, where vehicles move autonomously in a shared environment.

References

[1] Y. Chaaban, J. Hähner, and C. Müller-Schloer. "Measuring Robustness in Hybrid Central/Self-Organising Multi-Agent Systems". In Cognitive12: proceedings of the Fourth International Conference on Advanced Cognitive Technologies & Applications, July 2012, pp. 133-138, Nice, France.

[2] CAS-wiki: Organic Computing. http://wiki.cas-group.net/index.php?title=Organic_Computing, [retrieved: June, 2013].

[3] C. Müller-Schloer, H. Schmeck, and T. Ungerer. "Organic Computing — A Paradigm Shift for Complex Systems". Birkhäuser, Verlag 2011.

[4] J.O. Kephart and D.M. Chess. "The vision of autonomic computing". IEEE Comput. 1, 2003, pp. 41-50.

[5] R. Sterritt. "Autonomic Computing". Innov. Syst. Softw. Eng. 1(1), 2005, pp. 79-88.

[6] M. Zeller. Fraunhofer Institute for Communication Systems ESK, Germany. IARIA Work Group Meeting: Autonomic and Autonomous, 1. PANEL ICAS, Topic: Robustness and Trust in Autonomic Systems. Panel @ ICAS 2010. The Sixth International Conference on Autonomic and Autonomous Systems ICAS 2010, March 7-13, 2010 - Cancun, Mexico. http://www.iaria.org/conferences2010/filesICAS10/ICAS_2010_Panel.pdf

[7] J.P. Steghöfer, R. Kiefhaber, K. Leichtenstern, Y. Bernard, L. Klejnowski, W. Reif, T. Ungerer, E. André, J. Hähner, and C. Müller-Schloer, "Trustworthy Organic Computing Systems: Challenges and Perspectives", Proceedings of the 7th International Conference on Autonomic and Trusted Computing (ATC 2010), Springer.

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

[9] C. 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, 2004, pp. 2-5. ACM.

[10] U. Richter, M. Mnif, J. Branke, C. Müller-Schloer, and H. 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), 2006, pp. 112-119. Bonner Köllen Verlag.

[11] Y. Chaaban, J. Hähner, and C. 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.

[12] Y. Chaaban, J. Hähner, and C. 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, January 2010, pp. 341-346, Valencia, Spain.

[13] Y. Chaaban and C. Müller-Schloer. "Specifying Desired Behaviour in Hybrid Central/Self-Organising Multi-Agent Systems". In Cognitive13: proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications, May 27 - June 1, 2013, pp. 1-6, Valencia, Spain.

[14] Y. Chaaban, J. Hähner, and C. Müller-Schloer. "Handling of Deviations from Desired Behaviour in Hybrid Central/Self-Organising Multi-Agent Systems". In Cognitive12: proceedings of the Fourth International Conference on Advanced Cognitive Technologies and Applications, July 2012, pp. 122-128, Nice, France.

[15] Y. Chaaban and C. Müller-Schloer. "A Survey of Robustness in Multi-Agent Systems". In Cognitive13: proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications, May 27 - June 1, 2013, pp. 7-13, Valencia, Spain.

[16] K. Dresner and P. 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.

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

[18] 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.

[19] 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 2005, pp. 24-27.

[20] K. Waldschmidt and M. Damm. “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.

[21] W. Heupke, C. Grimm, and K. Waldschmidt. “Modeling uncertainty in nonlinear analog systems with affine arithmetic”. Advances in Specification and Design Languages for SOCs Selected Contributions from FDL ’05, 2005.