A Survey of Robustness in Multi-Agent Systems

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

***Abstract*—Nowadays, robustness is one of the several concepts that have to be considered when designing multi-agent systems. Thus, achieving robustness in multi-agent systems is of central importance. However, there is a clear lack of study of robustness in developing robust multi-agent systems in technical systems. In this paper, we provide a survey on robustness issues and mechanisms in multi-agent systems in diverse research fields. Afterwards, we suggest our interdisciplinary methodology, "Robust Multi-Agent System" (RobustMAS) to characterise robustness of multi-agent systems. RobustMAS poses a challenge to support the multi-agent system with mechanisms to keep the system at a desired performance level when disturbances and deviations from plan occur (robustness). Furthermore, RobustMAS proposes a new appropriate method to measure the robustness of such multi-agent systems.**

Keywords-Robustness; Survey; Multi-Agent Systems

I. INTRODUCTION

The ever increasing complexity of today's technical systems embodies a real challenge for their designers.

In this context, 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.

Although there are numerous research projects made towards building robust multi-agent systems in diverse fields, a study of robustness of technical systems, which are modelled as multi-agent systems, does not exist yet (at least it is extremely rare, e.g., an attempt by the Organic Computing Initiative [5]).

The next section gives an overview of existing related work aiming at highlighting the need of a novel approach to cover the gap recognised by designing robust multi-agent systems.

This paper is organised as follows. Section 2 presents a survey of related work concerning robustness of multi-agent systems in various research fields. In Section 3, the concept and objectives of RobustMAS will be presented. Additionally, the measurement of robustness of a multi-agent system according to the RobustMAS concept will be presented, where a new method for their measurement has been developed. Section 4 draws the conclusion of this work. Finally, the future work is explicated in Section 5.

II. STATE OF THE ART

In the literature, there are enormous works concerning the robustness of systems. However, there is a clear lack of study of robustness, to the best of our knowledge, in developing robust multi-agent systems in technical systems.

The development of robust multi-agent systems can address the robustness in the face of various kinds of factors (i.e., in the sense of turbulences) such as unreliable agents, faulty agents, malicious attacks, system uncertainty, common disruptions, failing elements or components, unreliable components, variable (turbulent) environments, environmental catastrophes, unexpected situations and exceptional conditions. In short, the goal is to develop a robust multi-agent system despite disturbances and deviations occurred in the system (intern) or in the environment (extern).

A. Robustness approaches in MAS

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 the literature. Additionally, various formal measures and metrics were presented to achieve the system robustness.

In the following, several research projects and approaches will be presented that are of interest in the context of this survey. They deal in some way with robustness of multi-agent systems in various research fields.

1) Handling communication exceptions in double auctions

Parsons and Klein [6] tried to build robust multi-agent systems against unreliable agents and infrastructures using a domain-independent exception handling approach. They have proved that their approach has the ability to achieve the robustness of multi-agent systems. Parsons and Klein provided every agent with a sentinel, so that each message from and to an agent can be processed (guarded) by such a sentinel. These sentinels are able to detect corrupted messages. Consequently, exception handling services, which are provided by sentinels, will enable robust multi-agent systems. A point of interest in this approach is that the exception handlers are generic. Therefore, the same exception handlers may be required for a broader variety of multi-agent systems [6].

In this regard, Parsons and Klein implemented their proposed approach for multi-agent systems that accomplish resource allocation using double auctions to handle communication exceptions.

2) The "citizen" approach

Similar to the work in [6], the so-called "citizen" approach, was presented by Klein et al. in [7]. The "citizen" approach tries to improve the robustness of multi-agent systems by off-loading exception handling from problem solving agents to distinct domain-independent services. It facilitates robust open multi-agent systems. This approach observes a multi-agent system in order to detect problems (exceptions) and consequently to intervene if needed. The case study of this approach was handling the agent death exception in the contract net protocol. According to the "citizen" approach, citizens embrace optimistic rules of behaviour but a whole host of social institutions will be used so as most exceptions can be handled (institutions deals with exceptions more efficiently than individual citizens). The main factor, which leads to applying the citizen approach efficiently to the development of multi-agent systems, is that widely reusable, domain independent exception handling expertise can be separated from the knowledge that agents in MAS can act upon to perform their usual jobs [7].

3) Agent Programming Language for Robustness (APLR): BDI agent programming

Another approach to support robustness of multi-agent systems was introduced by Unruh et al. in [8]. This approach is based on logging aiming to build more robust multi-agent systems. It tries to deal with problems occurring in multi-agent systems and consequently to recover from them. It uses an execution logging in order to build robust agents. The execution logging (execution history) has to be ensured at the architectural level. This means that agents in MAS should possess architectural-level support for logging and recovery methods when the robustness of MAS is considered. They presented also how an infrastructure-level logging approach can sustain agents so that run-time problems in BDI agents can be recovered [8]. Additionally, Unruh et al. have defined a special programming language, called APLR (Agent Programming Language for Robustness). This language is a developer-level language and defined especially for BDI agent programming. It aims to encode agent problem-handling knowledge so that a specification of problem-handling information will be supported as well as the developer can be insulated and constrained from the infrastructure-level reasoning [8].

4) Karlsruhe Robust Agent SHell (KRASH): In production planning and control (PPC)

In production planning and control (PPC), an approach by Frey et al. in [9] has addressed the robustness of such systems that were designed as multi-agent systems. In this context, flexibility and robustness are especially looked for in the case of production environments that are subject to continual, substantial and rapid changes in conditions (disturbances or turbulences). Frey et al. have applied database technologies on the basis of transactions in order to achieve the robustness of multi-agent systems. They assumed that robustness and reliability, which are common characteristics of current database systems, will solve the detected lack of reliability and robustness in the industrial deployment whether database technologies are applied. The database technologies will allow agents to perform their tasks robustly via providing robustness services, since robustness services are widespread in database techniques.

In this regard, Frey et al. have used transaction trees (common tree-like structures) to represent the agent plans. Next, execution agents, which are particular components of their multi-agent system, were assigned to execute the agent plans in a robust manner. So, transactions (transaction-based recovery mechanisms) ensure the recoverability in presence of disturbances [9].

Additionally, it is assumed that MAS can handle this problem more effectively than conventional centralised approaches on account of their flexible and robust behaviour. Frey et al. have modelled a multi-agent system and then compared it to an Operations Research Job-Shop algorithm. The comparison was made using a simulation-based benchmarking scenario. According to this approach, robustness on the shop floor will be assured by using MAS and rescheduling algorithms. As a result, robustness of a production system against disturbances can be supported not only by scheduling algorithms but also by a proper MAS architecture [10]. On the other hand, a simulation-based benchmarking platform was developed at the University of Karlsruhe in Germany. This platform was part of the Karlsruhe Robust Agent SHell (KRASH) project that is based on a real world production scenario (shop floor scenario). The goal of the benchmarking platform was to discover whether MAS can improve the planning quality in the shop floor scenario. In short, due to the fact that robustness is a significant aspect of a manufacturing system, this approach presented a transaction-based robustness service using database technology so that disturbances (e.g., machine failures) can be handled [9].

5) Transactional conversation: Layered agent implementation architecture

Closely related to the work in [9], an approach was developed by Nimis and Lockemann in [11] aiming to increase the robustness of multi-agent systems. This approach is called transactional conversation. It applies transaction-based robustness mechanisms, which are common in database management systems (DBMS). These mechanisms were integrated in a robust FIPA-compliant MAS development framework.

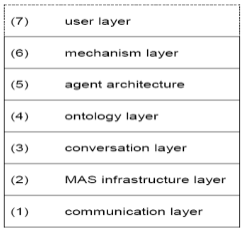

Figure 1. Layered agent implementation architecture [11]

More accurately, agent conversations will be handled as distributed transactions. Nimis and Lockemann have defined the robustness of MAS as following: “Robustness for Multi-Agent Systems means their ability to show predefined qualitative behavior in the presence of unaccounted types of events and technical disturbances”. According to this definition, the problems arising from disturbances during the agent interaction should be resolved so that the operation of MAS will be more robust. This approach was applied to applications of production planning and control (PPC). Nimis and Lockemann have presented an agent implementation architecture that was used as a framework to argue about the various aspects of robustness as well as to categorise the heterogeneous approaches in this area to increase the robustness of MAS. This architecture is a structured overview that organises the development tasks in ascending order based on abstraction levels that lead in turn to several layers (layered architecture) as depicted in Figure 1. In this regard, diverse issues were taken into account from the point of view of developers through the development process in order to build that agent architecture [11].

Obviously, in this work, the concentration of robustness considerations has to be on the third layer, the conversation layer, where the agent cooperation is controlled, because this layer is the most critical layer for ensuring the general robustness of MAS.

6) Robustness in Information Systems (IS): an underlying middleware

Similar to the works in [9][11], a promising approach was developed by Nagi in [12] demonstrating a first step towards achieving robust multi-agent systems. This approach aims to increase the robustness of a multi-agent system that is applied in the distributed Information Systems (IS) field of study by means of an underlying middleware. This middleware has to guarantee the robustness of the MAS. The main point in this interdisciplinary approach is to discuss the relation between the technologies of both agents and databases, where agents need to share data asynchronously. Thus, Nagi claims that the agents of a MAS share a world model in which the present situations can be reflected in a common database. Nagi has defined the robustness as follows: “We define the robustness provided by the middleware in terms of guarantees given on a technical basis, which is guaranteeing the correctness in normal operation and recoverability of the system in case of disturbances.” [12]. The key idea of this work is to develop an extended transaction model encompassing agent plans and their emergency behaviour (emergency behaviour in the case of disturbances in order to react to them). Additionally, an execution agent has to be involved in order to execute this transaction model. This execution agent ensures the robustness of execution of agent actions. At the same time, the execution agent characterises the interactions with different elements of a generic MAS architecture [12]. Figure 2 shows the proposed robust MAS architecture.

Here, the environment (the world) of the MAS is represented by databases. Every agent perceives its environment (reading from databases) and possibly changes it by producing certain actions (writing to databases). It is noteworthy that every agent is divided in two entities; a planning agent (located in the planning layer) and an execution agent (located in the execution layer) [13]. A planning agent has to cooperate with other planning agents to create the common shared plan, where each planning agent has its common goal and creates its local part of the shared plan and then hands it over to a peer execution agent. Every execution agent executes its received local plan, where coordination protocols will be used to coordinate the execution with other execution agents. In this context, local plans are represented by transaction trees, so that each single-agent action will be encapsulated in an ACID transaction (atomicity, consistency, isolation and durability). ACID transactions are utilities that can be used to guarantee the robust execution of agent actions, where it is known that the most commercial DBMS provide the ACID transactions [12].

7) Market-style open MAS: delegation concept, social agents

In the context of market-style open multi-agent systems, a study was introduced by Schillo et al. in [14] in order to define robustness quantitatively in such systems. This study assumes that robustness of MAS is more than redundancy, because problems caused by malicious agents in open systems can not be solved using redundancy. Therefore, it defines robustness of MAS with respect to performance measures and consequently a robust MAS keeps safety responsibilities despite events, which cause disturbances. Thus, a MAS system will be robust if it preserves a certain level of performance. Schillo et al. presented quantitative definition of robustness, using an electronic market, as follows: “the expected drop of the performance measure in four perturbation scenarios (i) increase of population size, (ii) change of task profile over time, (iii) malicious agent intrusion, and (iv) drop-outs of agents.” [14]. According to this definition, robustness of the MAS presents the amount of performance decrease measured in a perturbation scenario (e.g., in case of double population size). The contribution of this study lies in social agents, organisation of agent societies and robustness of social systems.

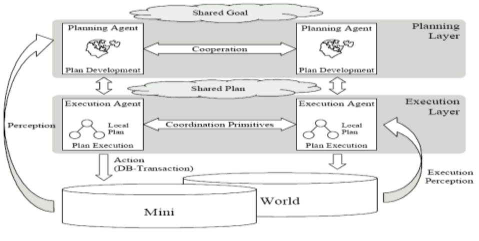

Figure 2. Proposed robust MAS architecture [13]

In short, Schillo et al. suppose that the four properties (scalability, flexibility, resistance, and drop-out safety), which are required to cope with the perturbation scenarios, will be accomplished using [14]:

1) Two types of operation (task delegation and social delegation).

2) Four mechanisms for delegation (voting, authority, economic exchange and social exchange).

It is important to pay attention that delegation is a complex concept that is very significant in the context of MAS. The delegation concept facilitates attaining robustness and flexibility of MAS. Task delegation is based on delegating tasks to other agents, which leads in turn to agents specialising in certain tasks [14].

8) Delegation concept: Holons, social order

Another related work to attain robustness of multi-agent systems using the delegation concept was introduced by Schillo et al. in [15]. This work is based on simulation of social systems using the "social order" concept in the social sciences, because social order bears similarity to robustness in this context. Additionally, it illustrates the properties that agents should have in order to develop them in complex social systems. In this regard, the concept of flexible holons was used. This concept depends on arrangement of agents in groups (task delegation and social delegation) to model institutions in MAS and consequently to utilise their facility of achieving robustness of MAS. Thus, Schillo et al. have analysed the delegation between agents and applied it to holonic systems. Holons (holonic agents) are a useful method for purposes of modelling institutions in MAS [15]. Here, a holon (a holonic agent) consists of parts, which in turn are agents. As a result, a holonic agent is part of a whole and consequently it assists to attain the aims of this superior whole. Additionally, modelling of institutions will make MAS robust, since institutions reduce complexity. A dynamic electronic market, which is able to manage transportation orders, has served as scenario for this work, where agents were created for this purpose [15].

9) Self-Organization and Robustness in Multi-agent systems (FORM)

One additional study was performed by Schillo et al. in [16], which proposed a new sociological concept. It studied self-organisation in multi-agent systems. Of particular interest in this study is the developed Framework for Self-Organization and Robustness in Multi-agent systems (FORM). The reason for that is that robustness (within the meaning of scalability) is closely related to self-organisation in some application scenarios. Schillo et al. have illustrated this framework with respect to the sociological features of organisations. FORM characterises organisational forms and relationships by means of the delegation concept in MAS. In short, the FORM-framework is used to model (and hereby to accomplish) self-organisation of MAS organisations [16].

10) Organisational forms of MAS: genetic algorithms

Most closely related to the work in [16], a new concept was introduced by Hahn et al. in [17] investigating organisational forms of MAS. This concept aims to build robust MAS utilising genetic algorithms that can be used as a search heuristic, since genetic algorithms are effective mechanisms to deal with enormous search spaces. Based on this, the implemented genetic algorithm searches this space for superior forms of organisation. Hahn et al. have defined robustness with respect to a performance measure as follows: "Robustness is considered as graceful degradation of a system's performance under perturbation." [17]. Therefore, in order to evaluate the performance of the recent discovered forms (will be formed by recombination of mechanisms) of organisation under various circumstances, diverse robustness criteria were defined according to sociological theory (for details see [17]). That means, the evaluation of those forms of organisation was based on their involvement (beneficial effects) in the performance and robustness of the MAS in order to search for optimal combinations of the mechanisms. On the other hand, organisational forms (structures) are characterised by the specific applied mechanisms. This means that the behaviour of each organisational form will develop via the different possible mechanisms used by the organisations to satisfy their particular attributes. Here, the numbers within the gene stand for the used mechanism. For example, the first gene characterises the mechanism used for task delegation, where three specific mechanisms were implemented (economic exchange, economic exchange combined with gift exchange or authority). Based on this, the search process delivers organisational forms that have the ability to conform and act in a certain way, so that they demonstrate the best possible performance [17].

B. Summary: Robustness in MAS

Many research projects in the area of multi-agent systems focus on robustness. These works investigate the robustness in various research fields, such as distributed Information Systems (IS), database technologies, social systems and organisation of agent societies. However, there is a clear lack of study of robustness in building robust technical systems, which are modelled as multi-agent systems.

It should be pointed out that only some of the aspects of the topic under study in this paper could be taken into account and therefore they were included in the survey. This can be traced back to the vastness of the topic. Therefore, the survey is concerned with some literature on robustness in multi-agent systems as found in a variety of fields.

There is also much other related work that can be found in the literature, e.g., on re-planning, plan repair, Teleo-Reactive (TR) behaviours [18], formal analysis of protocols for emergent behaviours, and so on. Finally, sensor networks can be considered as MAS and there is much research published on robustness and fault-tolerance in sensor networks (see [19] for an example).

III. THE ROBUSTMAS APPROACH

A. A concise introduction

The Organic Computing initiative [5] aims at building 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 the interdisciplinary methodology, RobustMAS, is to investigate the robustness of coordination mechanisms for multi-agent systems in the context of OC.

The next sections give an overview of the RobustMAS concept. It is a novel approach aiming to cover the gap, building robust technical multi-agent systems, recognised in the previous section.

B. An application scenario for RobustMAS concept

The application scenario used by RobustMAS is a traffic intersection without physical traffic lights. For this reason, an intersection control algorithm based on virtual traffic lights is used. Such scenarios contain and assemble the required concerns that can be used to build robust multi-agent systems. In this context, autonomous agents are autonomous vehicles, and the controller of the intersection is the central unit. However, the basic idea of the RobustMAS concept is applicable for other systems as well.

C. Generalisation of the RobustMAS concept

For generalisation of the RobustMAS concept, the current scenario used in this work, intersections without traffic lights, can be replaced also with the more general scenario, shared spaces. This generalisation may be possible due to the similarities between the working circumstances and the environments presented in both systems. In this regard, both systems can be considered as unregulated traffic space, where vehicles move in a fully autonomous way without traffic lights.

The general problem domain of RobustMAS is the resource sharing conflict (resource allocation problem). This is a dynamic coordination problem. RobustMAS tries to solve the question how agents move reliably in a common environment. Here, agents compete for the shared environment (shared resources) in order to move over it quickly, and coordination of these agents in their common environment has to be achieved. In order to avoid a potential resource sharing conflict in such multi-agent systems, RobustMAS introduces a coordination mechanism. This coordination mechanism is based on the idea of path planning (planning of resource allocation over a certain period of time), which must be performed taking into consideration other agents and the geometry of the shared environment in the configuration space-time (x, y, t).

However, the resource allocation in RobustMAS is characterised by “Spatial-dependent resource assignment”. Spatial-dependent resource assignment is a plan-based resource allocation in the 3-dimensional configuration space-time (x, y, t), so that the next requested resource at the next time-step is nearby (successive time-steps). That means if the space (x_1, y_1) is the allocated resource at the time-step (t_1) for an agent, then the planning algorithm must take into account that the next potential resource, the space (x_2, y_2), at the next time-step (t_2) for this agent has to be close (1-neighbourhood) to the previous allocated resource. In the same way, the next space (x_3, y_3), at the next time-step (t_3) for this agent has to be close to the former allocated resource, etc.

D. The RobustMAS concept

For the explanation of the RobustMAS concept, the words agent and vehicle are used interchangeably. Also, the term “shared environment” is used interchangeably for “centre of the intersection”.

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. In this regard, the normal state represents the system performance level at its best when no disturbances occur (under normal operating conditions). RobustMAS proposes a hybrid architecture solving the conflict between a central unit (an observer and a controller) and decentralised autonomous agents.

The realisation of the three steps of the concept of RobustMAS (path planning, observation, controlling) has been introduced in our earlier papers [1][2][3][4]. First, RobustMAS concentrates on planning of the desired behaviour (trajectories) of agents (vehicles) in a shared environment (traffic intersection). Second, the observation process is designed to detect deviations from the planned trajectories (desired behaviour). Third, the controlling step concentrates on the control process of the system to cope with the occurred deviations from the planned trajectories or disturbances (accidents).

In this regard, we presented the desired system architecture in [1][2]. This architecture was an observer/controller (O/C) architecture adapted to the traffic scenario. The O/C architecture (an intersection manager) is responsible for coordinating tasks. It performs first a path planning to determine collision-free trajectories for the vehicles (central). Here, a trajectory represents the path of an agent only inside the shared environment. This path planning is given to vehicles as a recommendation. Here, 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.

In addition, an observation of compliance with these trajectories is done, since the vehicles are autonomous (decentralised and they are allowed to behave in a completely autonomous way) and thus deviations from the plan (planned trajectories) are possible. Here, different coordination and replanning mechanisms as well as the capability of the system to operate under real time conditions have been investigated.

Finally, the controlling step is performed. For this purpose, a decision maker was used. This decision maker will be activated when the controller gets a deviation message from the observer. Based on this, the controller algorithm was developed and discussed, followed by the actions table of the controller. The actions table was structured to achieve a desired strategy distinguishing between four different situations (a deviation, a disturbance: an accident, a high priority agent: an emergency-car, and above emergency-threshold). Based on this, the controller decides on an appropriate corrective intervention. In this way, the controller aims to allow the system to return after deviations or disturbances to its normal state with minimal central planning intervention.

In earlier papers, we extended our prototype implementation with the aim of making it capable of handling disturbances (accidents) in the system environment (intersection) in [4]. Furthermore, handling of deviations from planned (desired) behaviour was studied in [3].

In order to conceive the basic idea of RobustMAS, three cases of the system operation will be considered:

1. Operation without disturbance.
2. Operation with disturbance without intervention.
3. Operation with disturbance with intelligent intervention.

Figure 3 illustrates the main idea of this concept in establishing a robust system that tolerates faults, disturbances and deviations, which could be occurred in the system.

As depicted in Figure 3, the performance (e.g., throughput) of the system is at its best (i.e., equal to 1) when no disturbances occur. When a disturbance occurs, the system performance would begin to fall and probably it would become worse (deteriorate) over time, if no corrective intervention is taken in due time. In contrast, if the corrective intervention is intelligent and fast enough, the system performance should improve in the course of time when a disturbance occurs. This means that the system performance remains acceptable despite the occurrence of disturbance.

The aims of the RobustMAS concept have been achieved by using different test situations, handling of deviations from desired behaviour [3] and handling of disturbances [4]. These test situations were proposed to perform the evaluation with respect to the goal of the RobustMAS concept and to measure its robustness.

E. Robustness metrics

In the context of the RobustMAS concept, we developed an appropriate metric for the quantitative determination of the robustness. It is a new method to measure the robustness of hybrid multi-agent systems [4].

In this regard, the cumulative system performance, i.e., the cumulative throughput (#Agents) was used for determining the reduction of the performance (system throughput) of RobustMAS after disturbances (accidents) and deviations from the planned trajectories occur.

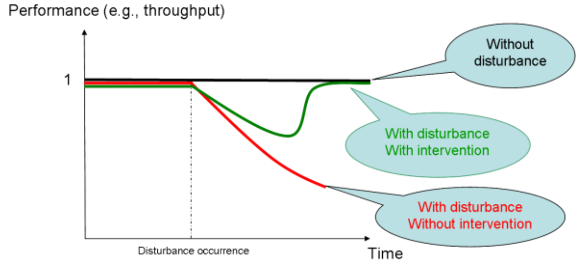

Figure 3. Robust system with disturbance occurrence

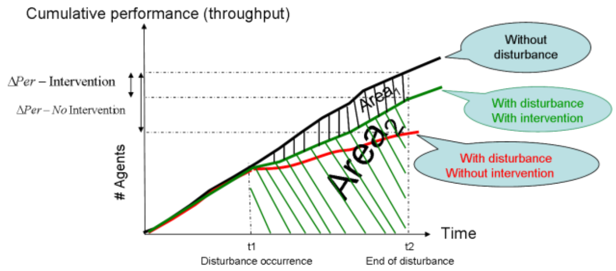

Figure 4. Measuring robustness using cumulative system performance

Here, the system is considered until the time when the disturbance ends. Therefore, the comparison of the throughput values is required in the three cases: without disturbance, with disturbance with intervention, and with disturbance without intervention (as depicted in Figure 4). It should be pointed out that the cumulative system throughput, #Agents, (the intersection throughput, #Vehicles) here is the total number of vehicles that left the intersection (simulation area) over time.

According to the RobustMAS concept, the relative robustness of a multi-agent system can be defined as follows: "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". Consequently, the relative robustness (R) of a system (S) can be determined as described in the next formula [4]:

$$R = \frac{\mathrm{Area}_2}{\mathrm{Area}_1 + \mathrm{Area}_2} = \frac{\int_{t1}^{t2} Per(t)_{(withIntervention)}\, d(t)}{\int_{t1}^{t2} Per(t)_{(NoDisturbance)}\, d(t)} \qquad (1)$$

Additionally, the discussion of the robustness measurement was 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 robustness values for the three cases were: 87%, 86%, and 83% respectively (for details see [4]).

However, the proposed robustness metric is not highly dependent on the application scenario studied by RobustMAS, a traffic system, and consequently it can be generalised to other application domains. In this regard, the general problem domain of RobustMAS, a resource allocation problem, can also be handled in a similar manner.

Here, agents consume the allocated resources. However, since the agents are allowed to behave in a completely autonomous way, they may agree to the allocation of resources (agree to the central plan) or they reject the allocation of resources (do not obey the central plan).

In the case of agent's consent to the plan, the resource allocation is optimal (no deviations from plan, the black curve), because the plan is performed by a central algorithm, which has a global view of all available resources that can be allocated to the system agents.

However, in the case of an agent's rejection of the plan, a potential resource allocation conflict between the agents is recognised, because of the consumption of resources, which are possibly reserved for other agents. When a conflict arises, it may be distinguished between two cases: (1) no new resource allocation is performed (the red curve) and (2) new resource allocation is made (the green curve).

IV. CONCLUSION

Robustness is a fundamental concern in multi-agent systems. The analysis of the state-of-the-art confirms that none of the addressed approaches or projects has filled the recognised gap, building robust technical systems, which are modelled as multi-agent systems. That means, there is no approach that is able to achieve such needed systems satisfactorily.

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. Therefore, RobustMAS supports the multi-agent system with mechanisms aiming to achieve the required robustness against disturbances. Additionally, it suggests a new method to measure the robustness of hybrid central/self-organising multi-agent systems.

Briefly, the RobustMAS concept raises the question how the robustness can be guaranteed and measured in technical systems. As a result, RobustMAS ensures a relatively acceptable level of reduction of the system robustness against increasing of disturbance strength.

V. FUTURE WORK

There are many interesting issues that can be explored beyond the mentioned above such as the extent of knowability. Furthermore, meta-cognition is a personal quest for insights and consequently it holds a potential for expanding to personalised models, community model, etc.

As mentioned above, the basic idea of the RobustMAS concept is applicable for other systems as well. 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. “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] 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, Spain.

[3] 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, France.

[4] 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, France.

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

[6] S. Parsons and M. Klein. “Towards Robust Multi-Agent Systems: Handling Communication Exceptions in Double Auctions”. Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems - Volume 3, July 2004, pp. 1482-1483, New York, USA.

[7] M. Klein, J. A. Rodriguez-Aguilar, and C. Dellarocas. “Using domain-independent exception handling services to enable robust open multi-agent systems: The case of agent death”. Journal of Autonomous Agents and Multi-Agent Systems, 7(1/2), July 2003, pp. 179-189.

[8] A. Unruh, J. Bailey, and K. Ramamohanarao. “A Logging-Based Approach for Building More Robust Multi-agent Systems”. Intelligent Agent Technology. IAT '06, vol., no., 18-22 Dec. 2006, pp. 342-349.

[9] D. Frey, J. Nimis, H. Wörn, and P. C. Lockemann. “Benchmarking and robust multi-agent-based production planning and control”. Engineering Applications of Artificial Intelligence In Intelligent Manufacturing, 2003, pp. 307-320.

[10] D. Frey and H. Wörn. “Das Benchmarking und der robuste Betrieb eines MAS in der Produktionsplanung und -steuerung” (in german). Institut für Prozessrechentechnik, Automation und Robotik; Universität Karlsruhe (TH), 2000.

[11] J. Nimis and P. C. Lockemann. “Robust Multi-Agent Systems: The Transactional Conversation Approach”. In: 1st International Workshop on Safety and Security in Multiagent Systems (SASEMAS04), July 2004, New York, USA.

[12] K. Nagi. “Transactional Agents: Towards A Robust Multi-Agent System”. Universität Karlsruhe, Ph.D., Lecture Notes in Computer Science, Springer, 2001.

[13] K. Nagi, J. Nimis, and P. C. Lockemann. “Transactional support for cooperation in multiagent-based information systems”. In: Proceedings of the Joint Conference on Distributed Information Systems on the basis of Objects, Components and Agents (VertIS), October 2001, pp. 177-191, Bamberg.

[14] M. Schillo, H. J. Bürckert, K. Fischer, and M. Klusch. “Towards a Definition of Robustness for Market-Style Open Multi-Agent Systems”. In: Proceedings of the Fifth International Conference on Autonomous Agents (AGENTS-01), Canada, 2001, pp. 75-76, ACM Press, New York.

[15] M. Schillo, I. Zinnikus, and K. Fischer. “Towards a Theory of Flexible Holons: Modelling Institutions for Making Multi-Agent Systems Robust”. In Proceedings of the Workshop on Norms and Institutions at Agents 2001, Montreal.

[16] M. Schillo, B. Fley, M. Florian, F. Hillebrandt, and D. Hinck. “Self-Organization in Multiagent Systems: From Agent Interaction to Agent Organization”. In: Proceedings of the Third International Workshop on Modelling Artificial Societies and Hybrid Organizations (MASHO'02), Aachen.

[17] C. Hahn, B. Fley, and M. Schillo. “Optimisation of Multiagent Organisation for Robustness”. In: Sozionik aktuell, Vol. 3, 2003, pp. 1-13.

[18] N.J. Nilsson. “Teleo-reactive programs for agent control”. Journal of Artificial Intelligence Research, 1, January 1994, pp. 139-158.

[19] S. Kiyomoto, K. Fukushima, and Y. Miyake. “Robustness Analysis for Sensor Networks against a Node-Destruction Attack”. IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.6, June 2012, pp. 10-17.