Maintaining Organizational Multi-agent Systems: A Reorganization-based Preventive Approach

Nawel Ghrieb[1,3], Farid Mokhati[2,3] and Tahar Guerram[2,3]
[1]*Department of Mathematics and Computer Science, University of Tebessa, Algeria*
[2]*Department of Mathematics and Computer Science, University of Oum El Bouaghi, Algeria*
[3]*RelaCS2Laboratory, University of Oum El Bouaghi, Algeria*

Keywords: Preventive Maintenance, OCMAS, Software Quality, Reorganization, AOP.

Abstract: In this article, we propose a preventive maintenance approach for Organization-Centered Multi-Agent System (OCMAS). This approach is based on the quality assessment for the maintenance of OCMAS. The quality of OCMAS is monitored by Aspect Oriented Programming (AOP) techniques, in order to detect any abnormal regression in the quality of the system or that of the agents composing it. This degradation in quality is, usually, an indication of problems that may arise in the structure of the organization or its functionalities. In the context of this work, we are interested in the functional problems that can affect the system and we treated them by reallocating agents to roles. This reallocation is, generally, necessary when the agent is not able to achieve its objectives, which leads to a degradation of the overall quality of the system. Our maintenance approach must therefore anticipate these problems and react by reorganizing the running system in order to improve its quality and allow it to resume its normal behaviour.

1 INTRODUCTION

Organizational multi-agent systems engineering has been proposed as a way to design complex systems that adapt to their environment. Agents interact and may assign tasks to each other based on their individual abilities (Ferber et al., 2004). However, the system designer may not have foreseen all possible environments in which the system can be deployed. Unpredictable application environments make multi-agent systems vulnerable to individual failures that can dramatically reduce the system's ability to accomplish its objectives. The agents themselves may exhibit particular properties that were not originally intended, and some agents may then become unable to fulfill the roles assigned to them. For example, dangers or even malicious actions could sever the communication links between agents. Alternatively, agents cannot fulfil their roles due, for example, to: insufficient capacities of agents to accomplish their roles or agent overload resulting in failure to meet some of its objectives.

As multi-agent systems grow, the configuration and tuning of these systems can become as complex as the problems they claim to solve. Thus, a robust organizational multi-agent system must be maintained in order to adapt to environments, recover from degradation in its performance (quality) and improve over time to avoid undesirable situations.

Therefore, the central question addressed in this article can be formulated as follows: «How to ensure the continuity of the MAS' activities so that the degradation of the performance (quality) of the agents in playing their roles does not drastically affect the functioning of the MAS? ». To address these concerns, we propose, in this paper, a preventive maintenance approach to enable organizational MAS to safely reach their goals. Preventive software maintenance is an important software activity that involves including changes and updates to prevent serious software problems in the future (ISO/IEC/ IEEE 24765, 2010). In the context of multi-agent systems, such activity is completely omitted. The inherent specificities to multi-agent systems (e.g. autonomy, pro-activity, reactivity, adaptability, etc.) make their maintenance difficult to achieve.

The proposed maintenance approach concerns, in particular organizational MAS. Wooldridge et al. (Wooldridge et al., 2000) view an organization as a collection of roles that stand in certain relationships

Ghrieb, N., Mokhati, F. and Guerram, T.
Maintaining Organizational Multi-agent Systems: A Reorganization-based Preventive Approach.
DOI: 10.5220/0010314803840389
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 1*, pages 384-389
ISBN: 978-989-758-484-8

to one another, and that take part in systematic institutionalized patterns of interactions with other roles. This type of MAS achieves its goals by assigning agents to different roles according to their individual abilities (Ferber et al., 2004). We then admit, in the context of this work, that an organization works properly and can achieve its objectives if its agents perform their roles in an efficient manner and that the decrease in their efficiency entails a disturbance which leads to a degradation of the efficiency of the entire system. Therefore, a solution where multi-agent systems are enhancing with the ability to reassign responsibilities from defective agents to others providing similar capabilities is needed. This requires first collecting information on the ability or inability of a given agent to assume its roles. This information can then form the basis of a decision as to whether other agents should be sought for the accomplishment of a given objective.

Based on these ascertainment, we propose a conditional preventive maintenance approach by monitoring certain quality criteria corresponding to the running MAS in order to predict any failure that may appear. The idea of our approach therefore consists in keeping the system's quality level and that of the agents beyond certain thresholds defined by the designer. When the system quality deteriorates below the defined thresholds, we reorganize the system in order to restore its quality and allow it to resume its normal functioning.

The remainder of this paper is organized as follows. In Section 2, we give a brief overview of major related work. We present in Section 3 the proposed approach. The architecture of the proposed system is given in section 4. We discuss in section 5 the advantages and limitations of our approach. Section 6 gives some conclusions and future work directions.

2 RELATED WORK

Very few approaches have been proposed in the literature to deal with the problems which are related to preventive maintenance of software (Garget et al. 1998, Vaidyanathan et al, 2002, Singh and BinduGoel, 2007, Cheluvaraju et al, 2012, Sun and Wang, 2012).

In order to solve the problem of preventive maintenance of software systems for transactions, an analytical model has been proposed by Garget and al. This model takes into account: the availability of the software to provide a service, the probability of losing a transaction and the response time of a transaction (Garget et al. 1998).

Through a Markov regeneration process with a subordinate semi-Markov reward process, Vaidyanathan et al. proposed an analytical model of a software system using preventive maintenance based on inspection (Vaidyanathan et al, 2002).

Singh and BinduGoel first attempted to analyse the issues governing software maintenance and how preventive maintenance can improve the lifespan (aging) of the software product. These authors then integrated a model for preventive maintenance into the software lifecycle (Singh and BinduGoel, 2007).

Cheluvaraju et al. proposed a software quality metric called "prevention metric" to measure the avoidance of defects in software. This metric derives from a quantitative assessment of both the efficiency and effectiveness of individual prevention techniques that are employed on the software prior to its deployment. It helps provide confidence in how defect prevention is handled prior to deployment (Cheluvaraju et al, 2012).

A preventive software maintenance policy based on the ant colony algorithm has been studied by Sun and Wang. The system as a whole has been divided into several subsystems each of which has four types of maintenance policy with different maintenance costs. This type of model makes it possible to obtain an optimal preventive maintenance policy for each sub-system, thus guaranteeing excellent reliability of the software system with relatively lower costs (Sun and Wang, 2012).

Ghrieb et al. (Ghrieb et al., 2020) have proposed a conditional preventive maintenance approach for multi-agent applications that is based on MAS quality measurements and uses aspect-oriented programming. This proposed approach includes three major steps: (i) measurements of two quality metrics (autonomy and sociability) of the running application in a dynamic and continuous manner using the AspectJ code and comparing them to the minimum thresholds previously defined by the designer, (ii) warning the maintainer in case of detection of abnormal regression in the MAS quality, and (iii) intervention by the maintainer to preserve the quality of the application and thus avoid potential damage.

As mentioned above, all of these approaches relate to preventive software maintenance. However, excepting the approach we proposed in (Ghrieb et al., 2020), none of this work deals with preventive maintenance of MAS. Our proposed approach was a first step towards proposing a generic preventive maintenance approach for multi-agent applications.

In this article, we present a new and original approach to conditional preventive maintenance for Organizational MAS. The proposed approach uses, on the one hand, aspect-oriented programming to monitor the quality of the running MAS and, on the other hand applies reorganization techniques to allow the system to resume its efficiency in case of problems.

3 THE PROPOSED APPROACH

The conditional preventive maintenance approach we propose is based on MAS quality measurements and uses aspect-oriented programming (Figure 1).

Our approach is used to continuously measure the quality metrics of the application and the agents composing it using the control code written in AspectJ and compare them with thresholds previously defined by the designer. When the measured values are lower of the specified thresholds, preventive intervention must be carried out in order to restore the quality of the agents to normal values above the previously defined thresholds. This intervention is carried out by reorganization techniques allowing the reallocation of agents to roles.

4 SYSTEM ARCHITECTURE

The overall architecture of the system implementing our approach is illustrated by figure 2. It is composed of two parts: our Maintenance System baptized Main-OCMAS (MAINtenance of Organization-Centered Multi-Agent Systems) and the Organization-Centered Multi-agent System (OCMAS), the running system which is composed of the Organizational Master (OM) and the agents of the application.

4.1 OCMAS System

The OCMAS system is designed according to a centralized approach (Figure 3), which makes it possible decoupling the reasoning part of the organization from the real system composed of the agents of the application. The Organization Master (OM) is the only agent with complete knowledge on the organization that is able to execute reorganization algorithms. The OM uses their knowledge on goals and current agents, makes the appropriate assignments, and sends the assignments to agents. The OM also receives events about the execution of each agent's objectives and reorganizes itself appropriately when needed. In this centralized approach we adopt, application agents are independent of any organizational model. They receive assignments from OM, play the roles assigned to them and report on their status to OM. Communication between the MO and the agents is done by sending messages.

4.2 Maintenance System MAIN-OCMAS

The MAIN-OCMAS maintenance system is an extension added to OCMAS system to provide condition-based preventive maintenance of OCMAS systems. This maintenance system continuously evaluates the OCMAS system in order to remedy any problems that may affect its organization and prevent it from achieving its goal. This system is composed of a Monitoring System (MS) and a Maintenance Agent (MA).

4.2.1 Monitoring System

The monitor runs as a third party and consists of gathering information using the Aspect Engine. The latter intercepts the relevant events of the agent platform: Communication monitoring, Message monitoring, Agent action monitoring in order to report all events requiring intervention to the Organizational Master (OM).

As our objective concerns the preventive maintenance of organizational multi-agent systems, so we are interested, in this work, in particular in the events leading to the regression of the efficiency of the system. These events are monitored by the monitoring system and will be used by the maintenance agent for the reorganization of the system in order to restore its efficiency value to the adequate level and thus prevent the system from a possible failure.

4.2.2 Maintenance Agent

The maintenance agent, in turn, is composed of an Evaluation Component (EC), a Disturbance Detection and Decision Component (DDDC) and a Reassignment Component (RC). The Monitoring system is responsible for retrieving the relevant information from the running MAS, the evaluation component assesses the effectiveness of the organization and that of the agents, the disturbance detection and decision component uses these

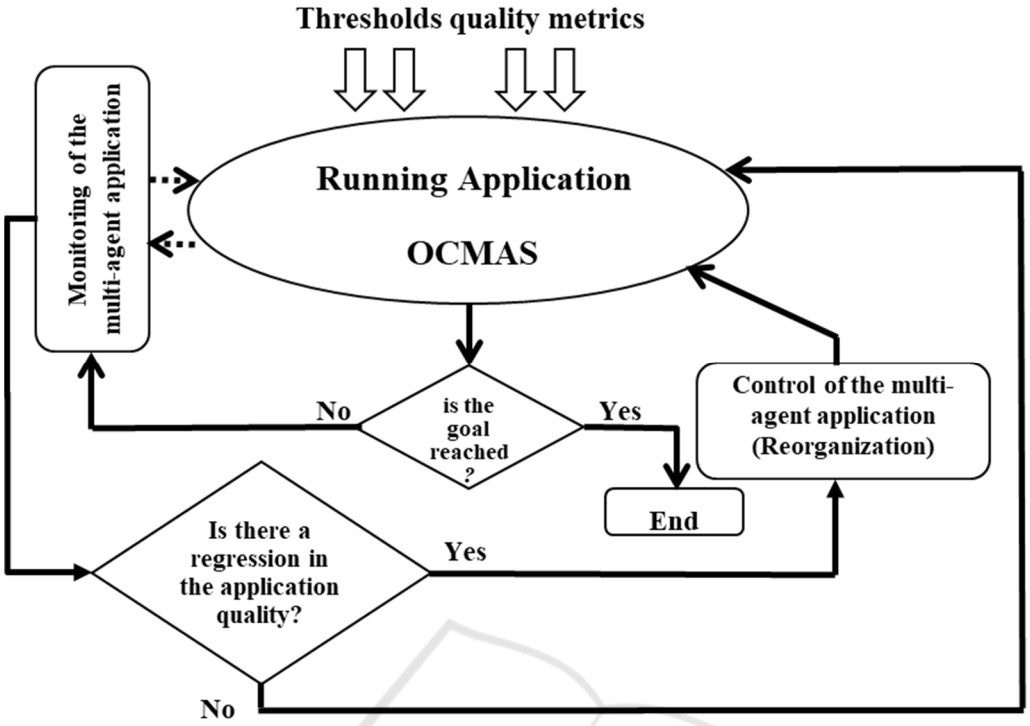

Figure 1: The methodology of the proposed approach.

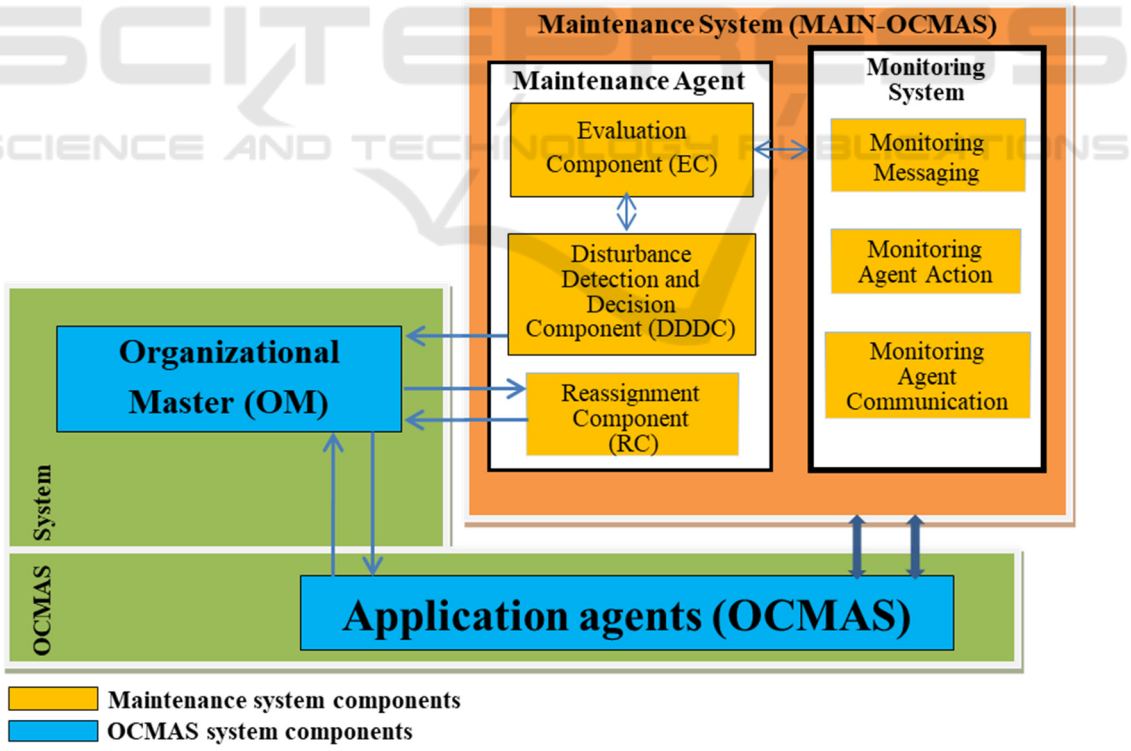

Figure 2: Overall system architecture.

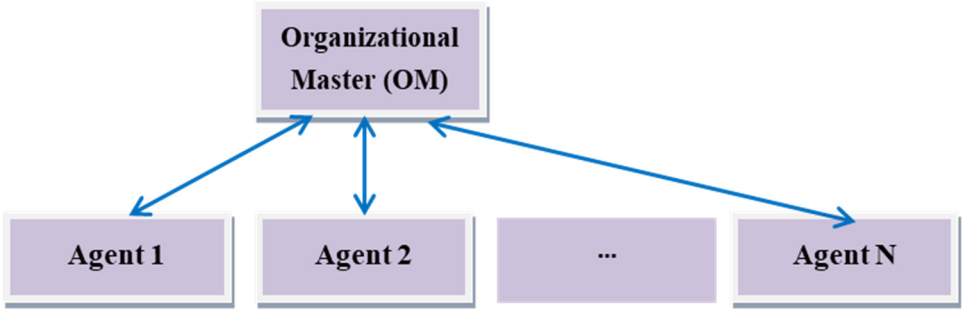

Figure 3: Centralized architecture of the OCMAS system.

assessments to identify situations in which: the quality requirements are not being met, the agents are not able to play their role well, and the current organization no longer meets the needs of the running MAS. The reassignment component reallocates agents to replace agents that do not perform their roles properly with more efficient agents. Finally, the organization master uses the reassignment component for the reorganization of the system, in order to improve the efficiency of the OCMAS system and allow it to reach its goal safely. The maintenance agent (figure 4) in our MAIN-OCMAS system is designed to ensure preventive maintenance of the OCMAS system; it accomplishes three main activities:

- **Perception.**

The maintenance agent watches for system disruptions from the execution trace established by the monitoring system. This component is constantly listening and allows the agent to analyze the result of the execution of roles by the agents and to keep track of the encountered disruptions.

- **Evaluation and Decision.**

The agent assesses these disruptions and decides when to start the reorganization process. The decision depends on the overall rate of disturbances, when the degree of efficiency in the system drops below a certain threshold, a global adaptation of it become necessary.

- **Execution.**

At this stage, the agent applies a reorganization algorithm which acts on the assignments of agents to different roles, in order to improve, at each step of the algorithm, the assignment of agents to roles.

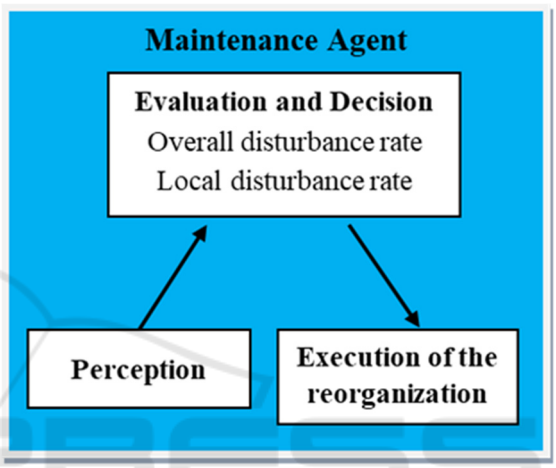

Figure 4: Model of the Maintenance Agent.

5 DISCUSSION

As noted above, some research works have been done on preventive software maintenance. These works have provided interesting solutions to different problems in different contexts. However, none of them deal with preventive maintenance of MAS. The approach we have already proposed in (Ghrieb et al., 2020), was a first step towards proposing a generic preventive maintenance approach for multi-agent applications. However, the limitation of this approach is its semi-automatic character. The approach presented in this article addresses this problem and allows automatic tuning of certain problems that may arise in organizational MAS. This approach is a new proposal that takes into account other quality attributes of multi-agent systems and allows preventive maintenance for organizational multi-agent applications. The maintenance system we have proposed in this article adopts a reorganization strategy that takes into account the drop in agent capacities and performs agent reassignment by assigning roles to the best

agents. This makes it possible, on the one hand, to improve the efficiency of the system in case of degradation of agent capacities and, on the other hand, to protect the system from harmful consequences in case of intrusion of non-competent agents into the running OCMAS. Furthermore, our monitoring system monitors the efficiency of the OCMAS system, in order to trigger a preventive intervention for preserving the system from an excessive drop in performance that could lead to its failure.

6 CONCLUSION AND FUTURE WORK

Multi-agent systems can become quite complex and can be deployed in open and unpredictable environments. The system designer may not have the resources to manually tune the system for a particular deployment. With these issues in mind, we propose a conditional preventive maintenance approach to create a self-tuning multi-agent system. The proposed approach is based on evaluating the assurance that OCMAS system agents are performing well to fulfill their corresponding roles. These performance metrics are used to trigger the automatic reassignment of roles to agents that is more effective, for example, in case of a danger which can lead to agent isolation or inability to achieve a goal. The reorganization currently applied by our MAIN-OCMAS maintenance system is achieved by a strategy of reassigning agents to roles in order to remedy the problems of the current organization. However, the architecture of our maintenance system is extensible; it allows adding other reorganization strategies in the maintenance agent in order to deal with other organizational problems. As future work, we plan to develop a software tool supporting our approach and allowing users to specializing it by offering them the possibility of choosing their organizational model.

REFERENCES

Ferber, J., Gutknecht, O., & Michel, F., (2004). From agents to organizations: an organizational view of multi-agent systems. In *Agent-Oriented Software Engineering IV*, Springer Berlin Heidelberg, pages 214–230.

ISO/IEC/ IEEE 24765, (2010), Systems and software engineering – Vocabulary.

Wooldridge, M., Jennings, N. R., David, K., (2000). The Gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems 3*: 285-312.

Garg, S., Puliafito, A., Telek, M. and Trivedi, K. S., (1998). Analysis of preventive maintenance in transactions based software systems. *IEEE Trans. Computers* 47(1): 96–107.

Vaidyanathan, K., Dharmaraja, S. and Trivedi,K.S., (2002). Analysis of Inspection-Based Preventive Maintenance in Operational Software Systems. In *Proceedings of 21st IEEE Symposium on Reliable Distributed Systems*, pages 286–295.

Singh, Y. and Goel, B., (2007). A step towards software preventive maintenance. In *Proceedings of ACM SIGSOFT Software Engineering Notes* 32(4):

Cheluvaraju, B., Pasala, A., Padmanabhuni,S., and Chevireddy,S.,(2012). A quantitative measure for preventive maintenance in software. In *ACM SIGSOFT Software Engineering*, Notes 37(4), pages 1–5.

Sun, P. and Wang, X., (2012). Application of ant colony optimization in preventive software maintenance policy. In *Proceedings of IEEE International Conference on Information Science and Technology*, pages 141–144.

Ghrieb, N., Mokhati, F., Ghorab, M. A., and Guerram, T., (2020). Towards a preventive maintenance approach for multi-agent applications. In *Multi-agent and Grid Systems* – An International Journal 16, pages 83–99.