Performance Evaluation of a Multi-Agent System using Fuzzy Model

Sabah Aly Darweesh Aly
Department of Computer and System Engineering
Faculty of Engineering, Ain Shams University
Cairo, Egypt
sabahaly289@yahoo.com

Hassan Mohamed Shehata Badoor
Department of Computer and System Engineering
Faculty of Engineering, Ain Shams University
Cairo, Egypt
hassan_bedour@eng.asu.edu.eg

***Abstract* – The Multi-Agent System performance evaluation became important because the development system using the Multi-Agent System became widespread. This paper evaluates the Multi-Agent System performance by evaluating each agent performance and then gets the overall Multi-Agent System performance. The implemented performance criteria are the intelligence, security and scalability criterion. We measure these criteria using the Goal/Question/Metric model, average function and Fuzzy Logic. Then, we generate the performance percentage using Fuzzy Logic system. Finally, we apply the suggested model on a sample of a Multi-Agent System (Code Advisory System), and analyse the result. The results are percentages of each criterion and the performance. In addition, we check the suggested model sensitivity if some entities of the model criteria are neglected.**

Keywords – Performance Evaluation of a Multi-Agents System Fuzzy Model, Goal-Question-Metric, Multi-Agent System

I. Introduction

Any Multi-Agent System (MAS) should perform as soon as the customer expectations. Can we measure the MAS performance as a value or a percentage? Can the agent performance criteria be determined? Can the criteria be measured? The work of this paper tries to answer these questions. We will introduce a model to determine the performance evaluation of the MAS. We assume that the performance means that how the agent performs in its Multi-Agent System environment. The measured results are percentages of each performance criterion.

We will present the idea as follows. Section II will discuss the related work. Section III will introduce the Goal/Question/Metric model. Section IV will introduce the Multi-Agent System performance evaluation benefits and issues. Section V will introduce the Multi-Agent System performance evaluation criteria. Section VI will introduce the performance evaluation of a Multi-Agent System using Fuzzy Model. Section VII will discuss the case study. Section VIII will introduce a conclusion and a recommendation of this work. Finally, the acknowledgment of this paper is introduced.

II. Related Work

Some researchers introduced ideas to evaluate the Multi-Agent System. Some of them were interested in the evaluating of the Multi-Agent System technology for a specific system. Others went to the comparisons among several Multi-Agent System platforms. In addition, some researchers tried to introduce an evaluation method. We will present samples of these trials.

A. *Evaluate specific MAS*

In June 2004, a researcher introduced an evaluation of the Multi-Agent System platform for using it as an infrastructure for ubiquitous computing. The evaluation process used two criteria groups, a "must have criteria" and a "ranking criteria". The two groups included the mobility standards security heterogeneity and the scalability. The result of this evaluation was that the MAS platform was dismissed or accepting [1]. In 2006, some researchers introduced a performance evaluation of a system. They tested the response time in three cases. The result showed that the Multi-Agent System was better than the single agent or the client-server system [2]. In December 2007, some researchers introduced an evolution of the agent performance using increasing the number of agents. The result was that there was minimum number of agents according to the cost and the noisy agents [3]. In December 2016, some researchers introduced an evaluation to show the benefit of the MAS using in the smart control [4].

B. *Evaluate for comparisons among several Multi-Agent System Platforms*

In June 2002, some researchers introduced an agent platform evaluation and comparison. The result explained that the recommended platforms were Grasshopper then JADE and then Aglets [5]. In 2002, some researchers introduced a performance evaluation of Zeus, JADE, and Skeleton-Agent frameworks based on the answers per the response time. The result was that the ZEUS architecture had a better behavior when there were a few agents. JADE platform had a regular behavior. Skeleton-Agent had a better performance for the Web-Agents, which had complex agent communications [6]. In April, 2004, some researchers introduced scale-up and performance studies of three agent platforms. The result was that the JADE platform had the best performance and the TRYLLIAN platform had the worst performance [7]. In 2008, a researcher introduced an evaluation of agent platforms using some questions about the agents. The result of this comparison was that the JADE platform was the best agent platform [8]. In January 2006, some researchers introduced a comparison of the performance of the Grasshopper and JADE agent platforms according to the time and traffic parameters. The result was that the best agent platform was the Grasshopper [9]. In June 2007, some researchers introduced a comparison of mobile agent platforms. These platforms were Aglets, Voyager, Grasshopper, TRYLLIAN, JADE, TRACY, and SPRINGS. The result was that the comparison could help a developer to know which platform was better [10]. In August 2010, some researchers introduced a performance evaluation of three multi-agent platforms (JADE, Mad-kit, and Agent-SCAPE). The result was that the JADE platform was the best [11].

C. Evaluation Methods

In 1991, a researcher introduced a definition of intelligence. This definition focused on whether the intelligent system achieved some performance goals such as success criteria [12]. In 2002, The MAS performance measurement was a set of questionnaires. These questionnaires were represented by a statistical approach or by fuzzy quantification [13]. In 2010, the intelligence metric was represented using the Goal\Question\Metric approach [14]. In 2015, The Z-Notation was used to evaluate the Multi-Agent System [15].

In this paper, it is the first time to introduce a general model to evaluate a Multi-Agent System's performance and test the model sensitivity. This model is called the Performance Evaluation of a Multi-Agent System using Fuzzy Model (PEMASFM) .We will use the intelligence, security and scalability criterion. This model is based on Fuzzy model and the Goal\Question\Metric (GQM) model. The next section explains the concept of the GQM model.

III. THE GOAL/QUESTION/METRIC MODEL

The Goal/Question/Metric (GQM) model is a model to determine the measurements of a software using three levels; the goal level (concept), the question level (operation) and the metric level (quantity) [16]. Each goal has some questions, each question has some metrics and each metric can have a value. The goal can share questions of another goal. The question can share metrics of another question. In this paper, we will use the GQM model to represent the performance criteria structure.

IV. THE MULTI-AGENT SYSTEM PERFORMANCE EVALUATION BENEFITS AND ISSUES

The performance evaluation of a multi-agent system has some benefits. We can have performance evaluation reports that play role in the improvement and management process. Despite the MAS importance, the MAS performance evaluation methodologies are very lack until now. The main reasons of this lack are the wide variety of agent functionality, the variety computational methods of the agent functions handling and the variety of MAS environments.

V. THE MULTI-AGENT SYSTEM PERFORMANCE EVALUATION CRITERIA

The MAS performance criteria are the intelligence, security, and scalability criterion. The MAS performance measurements will be a normal value or Fuzzy quantification value. Each criterion has a GQM model.

A. The Intelligence Criterion

We assume that the GQM model for the intelligence criterion has four goals. They are adapting, rationality, autonomy, and reactivity goal. We drive this criterion from the introduced work in 2010 [14]. The intelligence criterion includes seventeen metrics. Each question of intelligence goals is the average value of its metrics. Each goal of intelligence criterion is the Fuzzy Logic Controller output of its questions. Finally, the value of the agent intelligence criterion is the Fuzzy Logic Controller output of its four goals.

B. The Security Criterion

We generate a GQM model of the security criterion from some security concepts [17]. Table I introduces the GQM model of the security criterion.

TABLE I: THE GQM MODEL OF THE SECURITY CRITERION

The security criterion
Goal 1: Security Requirements
Question 1: Is the confidentiality existed?
M18 is the agent state privacy metric. It has a value of 1 when an unauthorized entity cannot observe the agent state and it has a value of 0 otherwise.
M19 is the agent code privacy metric. It has a value of 1 when an unauthorized entity cannot observe the agent code and it has a value of 0 otherwise.
M20 is the agent anonymity metric. It has a value of 1 when the agent is cannot be identifiable and it has a value of 0 otherwise.
M21 is the limited host data privacy metric. It has a value of 1 when parts of the host data privacy can be limited and it has a value of 0 otherwise.
M22 is the verifiable host data privacy metric. It has a value of 1 when parts of the host data privacy can be verified and it has a value of 0 otherwise.
M23 is the complete host data privacy metric. It has a value of 1 when parts of the host data privacy can be completed and it has a value of 0 otherwise.
Question 2: Is authentication existed?
M24 is the user authentication metric. It has a value of 1 when an authentication process authenticates the user and it has a value of 0 otherwise.
M25 is the host authentication metric. It has a value of 1 when an authentication process authenticates the host when the host communicates to a platform and it has a value of 0 otherwise.
M26 is the code authentication metric. It has a value of 1 when the host knows the purpose of the agent code and it has a value of 0 otherwise.
M27 is the agent authentication metric. It has a value of 1 when the host knows the owner of the agent and it has a value of 0 otherwise.
Question 3: Is integrity existed?
M28 is the agent state integrity metric. It has a value of 1 when the agent state should not be changed by a malicious entity and it has a value of 0 otherwise.
M29 is the agent code integrity metric. It has a value of 1 when the agent code should not be changed or not be run by a malicious entity and it has a value of 0 otherwise.
M30 is the server integrity metric. It has a value of 1 when the host data, state, and resources should not be changed by a malicious entity and it has a value of 0 otherwise.
Question 4: Is access control existed?
M31 is the access control of agent-to-agent platform metric. It has a value of 1 when the authorized agent can only work on a specific host of a platform and it has a value of 0 otherwise.
M32 is the access control of agent-to-host metric. It has a value of 1 when the agent can access resources in the host and it has a value of 0 otherwise.
M33 is the access control of host-to-agent metric. It has a value of 1 when the agent platform and trusted hardware are existed and it has a value of 0 otherwise.

The security criterion
M34 is the access control of agent-to-agent metric. It has a value of 1 when the agent can access the code, state, data or information of another agent and it has a value of 0 otherwise.
Question 5: Is non-repudiation existed?
M35 is the agent non-repudiation metric. It has a value of 1 when the agent should not wrongly assign its action to the server and it has a value of 0 otherwise.
M36 is the server non-repudiation metric. It has a value of 1 when the server should not wrongly assign the agent for action and it has a value of 0 otherwise.
Question 6: Is availability existed?
M37 is the agent accountability metric. It has a value of 1 when the agent must be accountable for its actions and it has a value of 0 otherwise.
M38 is the agent authorization metric. It has a value of 1 if the agent can deny service or can delay the resources using against the malicious entity and it has a value of 0 otherwise.
M39 is the agent code verification metric. It has a value of 1 when the agent code should be verified if the agent makes a critical job and it has a value of 0 otherwise.
M40 is the agent privilege revocation metric. It has a value of 1, if a malicious code or service attacks the agent, the agent privileges revocation are held and it has a value of 0 otherwise.
M41 is the host accountability metric. It has a value of 1 when the host must be accountable for its actions and it has a value of 0 otherwise.
M42 is the host availability metric. It has a value of 1 when the host data, resources and services can be accessed at any time and it has a value of 0 otherwise.
Goal 2: Avoiding Threats
Question 1: Is agent-to-platform threat avoided?
M43 is the agent-to-platform threats metric. It has a value of 1 when threats of an agent to a particular platform can be avoided and it has a value of 0 otherwise.
Question 2: Is platform-to-agent threat avoided?
M44 is the platform-to-agent threats metric. It has a value of 1 when threats of a platform to a particular agent can be avoided and it has a value of 0 otherwise.
Question 3: Is agent-to-agent threat avoided?
M45 is the agent-to-agent threats metric. It has a value of 1 when threats from the interaction among agents can be avoided and it has a value of 0 otherwise.
Question 4: Is platform-to-platform threat avoided?
M46 is the platform-to-platform threats metric. It has a value of 1 when threats among agent platforms can be avoided and it has a value of 0 otherwise.

Here, an avoiding is meaning that there is an avoiding process..

Each question of the security goals is the average value of its metrics. The first goal of security criterion is the average value of its questions. The second goal of security criterion is the Fuzzy Logic Controller output value of its questions. Finally, the value of the agent security criterion is the Fuzzy Logic Controller output of its two goals.

C. *The Scalability Criterion*

Finally, we generate a GQM model of the scalability criterion from the scalability concepts according to the agent technology. Table II shows the GQM model of the scalability criterion.

TABLE II: THE GQM MODEL OF THE SCALABILITY CRITERION.

The scalability criterion
Goal-1 is Adapting: This goal is the same as in the intelligence criterion [14].
Goal -2: Scale-Up/Scale-Down The metric value is 1 if the question answer is "Yes" and 0 otherwise.
Question 1: Can the agent be scaled up or scaled down on a platform?
M47: Can the agent be scaled up or scaled down on a platform?
Question 2: Can the agent be scaled up or scaled down across multiple platforms?
M48: Can the agent be scaled up or scaled down across multiple platforms?
Question 3: Can the agent knowledge be scaled up or scaled down?
M49: Can the agent knowledge be scaled up or scaled down?
Question 4: Can the agent types are scaled up or scaled down?
M50: Can the agent types are scaled up or scaled down?
Question 5: Can the agent task types are scaled up or scaled down?
M51: Can the agent task types are scaled up or scaled down?
Question 6: Can the agent task action types are scaled up or scaled down?
M52: Can the agent task action types are scaled up or scaled down?
Question 7: Are not more resources needed when the scaling up or the scaling down occurs?
M53: Are not more resources needed when the scaling up or the scaling down occurs?

The first goal of the scalability criterion is the first goal of the intelligence criterion. The second goal of the scalability criterion is the average value of its questions. Each question of the second goal has its metric value. Finally, the value of the agent scalability criterion is the Fuzzy Logic Controller output of its two goals. There are some Fuzzy Logic Controllers. We will explain it in details in the next section.

The intelligence, security and scalability criteria of a Multi-Agent System are the mean value of the intelligence, security and scalability criteria of all its agents respectively. The overall Multi-Agent System performance is the mean value of the performance of all its agents.

VI. THE PERFORMANCE EVALUATION OF A MULTI-AGENT SYSTEM USING FUZZY MODEL

A. *The PEMASFM Architecture*

The Performance Evaluation of Multi-Agent System Fuzzy Model architecture is illustrated in Fig. 1.

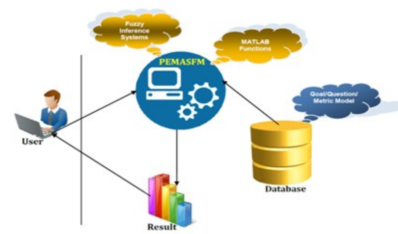

Fig. 1 The PEMASFM architecture.

- **The user** determines the metrics values and gets the model results.
- **The Database** contains the Goal/Question/Metric Model of the MAS performance criteria which is explained in the previous section. In addition, the database contains the function types for the calculation of question, goal, and criterion. There are three types of the function types; Fuzzy Logic, mean, and same function. Fig. 2 shows the database model of the PEMASFM model.

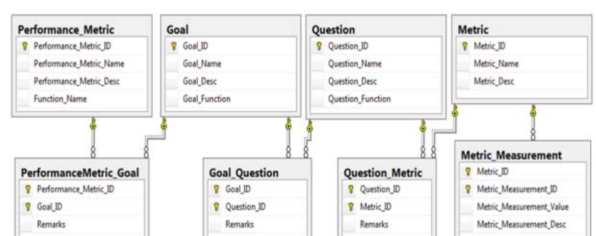

Fig. 2 The PEMASFM database model.

- **The MATLAB Fuzzy Inference Systems and Functions**: There are nine Fuzzy Inference Systems and the same number of the MATLAB functions to run it.
- **The PEMASFM:** When the user determines the metrics values, the PEMASFM calculates each entity in the GQM model using the function type, the MATLAB Fuzzy Inference Systems and its functions. Finally, the PEMASFM emerges the results as two bar charts. These charts contain each goal, each performance criterion and the performance percentage of the agent.

The PEMASFM tools are Microsoft Visual Studio, MATLAB and Microsoft SQL Server.

B. *The PEMASFM MATLAB Controllers*

The PEMASFM working is illustrated in the following controllers using MATLAB.

1) *Performance Evaluation Fuzzy Logic Controller*

The performance evaluation Fuzzy Logic Controller (FLC) is illustrated in Fig. 3.

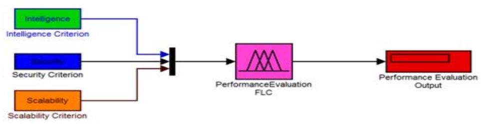

Fig. 3 The MAS performance evaluation FLC.

The Fuzzy rules of the performance evaluation FLC are illustrated in Table III.

TABLE III
THE FUZZY RULES OF THE PERFORMANCE EVALUATION FLC

Intelligence	Security	Scalability	Performance
Small	-	-	**Small**
Medium	Small	Small	**Small**
Medium	High	High	**Medium**
Medium	Medium	-	**Medium**
Medium	-	Medium	**Medium**
Medium	High	-	**High**
Medium	-	High	**High**
Medium	Small	-	**Medium**
Medium	-	Small	**Medium**
High	Medium	-	**High**
High	-	Medium	**High**
High	Small	-	**Medium**
High	-	Small	**Medium**
High	High	-	**High**
High	-	High	**High**
High	High	High	**High**

2) *Intelligence Criterion Fuzzy Logic Controller*

The intelligence criterion FLC is illustrated in Fig. 4.

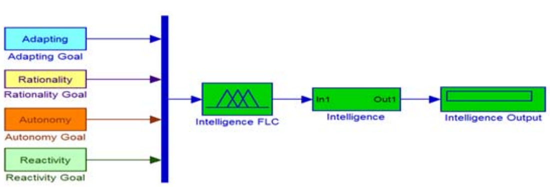

Fig. 4 Intelligence criterion FLC.

Some of the Fuzzy rules of the intelligence criterion FLC are illustrated in Table IV.

TABLE IV: THE FUZZY RULES OF THE INTELLIGENT CRITERION FLC.

Adapting Goal	Rationality Goal	Autonomy Goal	Reactivity Goal	Intelligence Criterion
Small	Small	Small	-	**Small**
Small	Small	-	Small	**Small**
Small	-	Small	Small	**Small**
-	Small	Small	Small	**Small**
Medium	Medium	Small	Small	**Small**
...	...	...	...	**...**

The membership functions of the adapting goal are illustrated in Fig. 5.

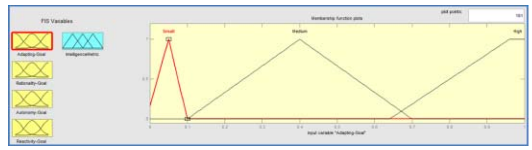

Fig. 5 The adapting goal membership functions.

a) *Adapting Goal Fuzzy Logic Controller*

The Fuzzy rules of the adapting goal FLC are illustrated in Table V.

TABLE V
THE FUZZY RULES OF THE ADAPTING GOAL FLC

Adapting Question 1	Adapting Question 2	Adapting Goal
Small	Small	**Small**
Small	High	**Medium**
Small	Medium	**Small**
Medium	Small	**Small**
Medium	Medium	**Medium**
Medium	High	**High**
High	Small	**Medium**
High	Medium	**High**
High	High	**High**

The Fuzzy Inference System of the adapting goal using the MATLAB Fuzzy tool is illustrated in Fig. 6.

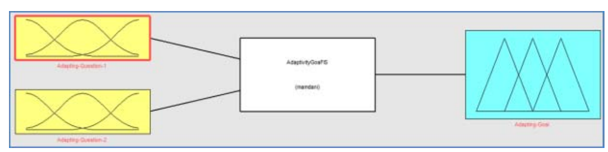

Fig. 6 The Fuzzy Inference System (FIS) of adapting goal.

The membership functions of the first question of the adapting goal are illustrated in Fig. 7.

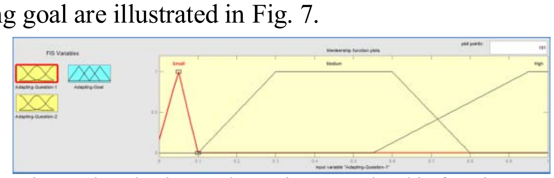

Fig. 7 The adapting goal question1 membership functions.

b) *Rationality Goal Fuzzy Logic Controller*

The rationality goal FLC is same as the adapting goal FLC.

c) *Autonomy Goal Fuzzy Logic Controller*

The autonomy goal FLC is same as the adapting goal FLC.

d) Reactivity Goal Fuzzy Logic Controller

The reactivity goal FLC is same as the adapting goal FLC.

3) Security Criterion Fuzzy Logic Controller

The security criterion FLC is shown in Fig. 8.

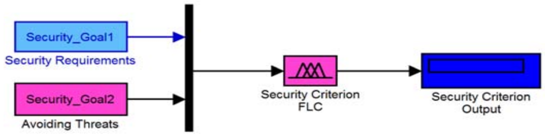

Fig. 8 The Security criterion FLC.

The security criterion FLC is same as the adapting goal FLC.

a) Security Requirements Goal Mean Controller

The value of the security requirements goal is the mean of its six questions.

b) Avoiding Threats Goal Fuzzy Logic Controller

The avoiding threats goal FLC is same as the intelligence criterion FLC.

4) Scalability Fuzzy Logic Controller

The scalability FLC is same as the adapting goal FLC. The second goal of the scalability criterion is the mean value of its seven questions.

C. The PEMASFM Interface

Fig. 9 shows the concept of the PEMASFM input interface.

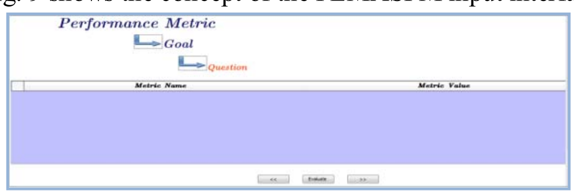

Fig. 9 The PEMASFM Interface Concept.

VII. CASE STUDY

A. The Code Advisory System

The Code Advisory System aims to improve the C# programmer efficiency [18]. The Code Advisory System is a Multi-Agent System. There is a dynamic way to create an agent for The Code Advisory System. The agent can have many roles. Each agent role has some tasks. Each agent task has some actions. The agent runs its action based on some business rules. The Code Advisory System has the Human-Interface, How-To-Do and What-Is-Wrong agents. The interface agent takes the request from the student, sends the student feedback and displays the response [18]. The response may be the response of the How-To-Do or What-Is-Wrong agent. The How-To-Do agent introduces some code snippets. What-Is-Wrong agent discovers the code errors. We apply the PEMASFM system on the Code Advisory System. The measured results are two bar charts and the percentages of the Code Advisory System performance as shown in Fig. 10 and Table VI.

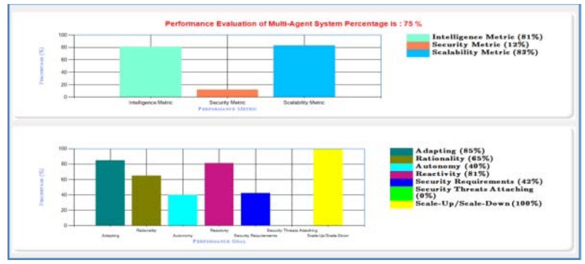

Fig. 10 The PEMASFM results of the Code Advisory System.

TABLE VI: THE PEMASFEM RESULTS OF THE CODE ADVISORY SYSTEM.

Results	Percentage
The Code Advisory System Performance	**75%**
Intelligence Criterion	**81%**
Security Criterion	**12%**
Scalability Criterion	**83%**
Adapting Goal	**85%**
Rationality Goal	**65%**
Autonomy Goal	**40%**
Reactivity Goal	**81%**
Security Requirements Goal	**42%**
Avoiding threats Goal	**0%**
Scale-Up/Scale-Down Goal	**100%**

B. Result Discussion

The percentage of the Code Advisory System performance is 75%. Where the intelligence percentage is 81%, the security percentage is 12% and the scalability percentage is 83%. The PEMASFM results of the Code Advisory System mean that the agent security is low. In addition, the intelligence might be higher if the learning metric and autonomy goal are existed.

C. Impact of the User Behavior on the result

We will try to check the impact of another user of the MAS team or the user behavior changing. These trials may be important to test the sensitivity of the PEMASFM model. The trial is a running of the PEMASFEM model with neglecting of some inputs. These trails results are shown in the Table VII.

TABLE VII: THE IMPACT OF THE USER BEHAVIOR ON THE CASE STUDY RESULT.

Neglecting	Results			
	Intelligence	Security	Scalability	Performance
Security criterion	81%	0%	83%	**69%**
Scalability criterion	81%	12%	40%	**75%**
Security & Scalability criteria	81%	0%	40%	**69%**
Autonomy Goal	57%	12%	83%	**70%**
Security's Goal 1	81%	0%	83%	**69%**
The basis of agent's pro-activity?	81%	12%	83%	**75%**
The agent has a role in the organizational structure?	57%	12%	83%	**70%**
The agent ability of each agent to perceive the environment?	54%	12%	83%	**71%**
Any question of the security requirements goal	81%	12%	83%	**75%**

If we compare the results in Table VI and the results in Table VII, we find the PEMASFM is more sensitive to the neglecting of the security or scalability criterion.

VIII. Conclusion

A. The PEMASFM Conclusion

The PEMASFM is a model that tries to measure the Multi-Agent System performance according to three criteria; intelligence, security and scalability criterion. Each evaluating criterion is based on the concept of Multi-Agent System using the Goal/Question/Metric model. Each criterion has some goals. Each goal has some questions. Each question has some metrics. The last level of the Goal/Question/Metric model contains the metrics which can be measured. The measurements of these metrics are either answers of questions or equivalent values to agent properties. Each upper level is calculated using an average, a same or Fuzzy function. The MAS manager, owner or developer can use the PEMASFM to evaluate his MAS performance. He can select one value for each metric. Finally the PEMASFM system evaluates the performance. The PEMASFM results are the goals percentages, the three criteria percentages and the performance percentage. The performance of the Multi-Agent System is the mean value of the performance of all its agents.

B. The Case Study Conclusion

The Code Advisory System is a Multi-Agent System that we run the PEMASFM to evaluate its performance. The PEMASFM result for the Code Advisory System performance is good for the intelligence aspect because the system has many characteristics of the MAS, but it is not satisfactory enough for the security aspect.

C. Recommendations

There are some recommendations of the PEMASFM.

- The suggested model can become a standard to evaluate the performance of a Multi-Agent System.
- The PEMASFM structure is a dynamic structure. New performance criteria can be added to improve the evaluation process.

Acknowledgment

This research is supported by Ain Shams University Faculty of Engineering, Computer Engineering and Systems Department as a task of a Master Thesis.

References

[1] Anders Liljedahl, "Evaluation of Multi-Agent Platforms for Ubiquitous Computing," Software Engineering and Computer Science, Blekinge Institute of Technology, Ronneby, Sweden, Dissertation MCS-2004-12, June 2004.

[2] Shahram Rahimi, Johan Bjursell , Marcin Paprzycki, Maria Cobb and Dia Ali, "Performance evaluation of SDIAGENT, a multi-agent system for distributed fuzzy geospatial data conflation," *Information Sciences*, vol. 176, no. 9, pp. 1175-1189, May 2006.

[3] Jerome Le Ny, Eric Feron, "Performance Evaluation of a Multi-Agent Risk-Sensitive Tracking System," in *46th IEEE Conference on Decision and Control*, New Orleans, 2007, pp. 2465-2469.

[4] R. K. Pandey, K. and Vamshi Kumar, "Multi Agent System Driven SSSC for ATC Enhancement," in *Power Systems Conference (NPSC), 2016 National: IEEE*, Bhubaneswar, India, 19-21 Dec. 2016, pp. 1-6.

[5] Nguyen G., Dang T.T, Hluchy L., Laclavik M., Balogh Z. and Budinska I., "Agent Platform Evaluation and Comparison," in *Pellucid 5FP IST-2001-34519*, Slovak, 2002, pp. 10-10.

[6] David Camacho, Ricardo Aler, Cesar Castro, Jose M. Molina, "Performance Evaluation of Zeus, Jade, and Skeleton Agent Frameworks," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 7, Tunisia, 2002, pp. 4-6.

[7] Simin Nadjm-Tehrani Kalle Burbeck, Daniel Garpe, "Scale-up and Performance Studies of Three Agent Platforms," in *Performance, Computing, and Communications: IEEE International Conference*, Phoenix, AZ, USA, USA, 2004, pp. 1-7.

[8] Rafał Leszczyna, "Evaluation of Agent Platforms," Institute for the Protection and Security of the Citizen (IPSC), Luxembourg, JRC Scientific and Technical Reports EUR 23508EN, 2008.

[9] Kresimir Jurasovic, G. Jezic, and Mario Kusek, "A performance analysis of multi-agent systems," *International Transactions on Systems Science and Applications*, vol. 1, no. 4, pp. 1-8, January 2006.

[10] Raquel Trillo, Sergio Ilarri and Eduardo Mena, "Comparison and Performance Evaluation of Mobile Agent Platforms," in *IEEE Third International Conference on Autonomic and Autonomous Systems: ICAS07*, Athens, Greece, 2006, pp. 41-41.

[11] Juan M. Alberola, Jose M. Such, AnaGarcia-Fornes, Agustin Espinosa, and Vicent Botti, "A Performance Evaluation of Three Multiagent," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 145–176, June 2010.

[12] James S. Albus, "Outline for a Theory of Intelligence," *TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, vol. 21, no. 3, pp. 473-509, MAY/JUNE 1991.

[13] Rong Gao and Lefteri H. Tsoukalas, "Performance Metrics for Intelligent Systems an Engineering Perspective. Measuring the Performance and Intelligence of Systems," in *Performance Metrics for Intelligent Systems :PerMIS'02*, Gaithersburg, MD, August 13 - 15, 2002, pp. 3-3.

[14] Pierpaolo Di Bitonto, Maria Laterza, Teresa Roselli and Veronica Rossano, "An Evaluation Method for Multi-Agent Systems," in *Agent and Multi-Agent Systems: Technologies and Applications: Springer Berlin Heidelberg*, vol. I, Gdynia, 2010, pp. 32–41.

[15] Christos Dimou, Fani Tzima, Andreas L. Symeonidis, and Pericles A. Mitkas, "Performance Evaluation of Agents and Multi-agent Systems Using Formal Specifications in Z Notation," in *International Workshop on Agents and Data Mining Interaction*, vol. 9145, Cao L. et al., 2015, pp. 64–78.

[16] IEEE Std 610.121990, "Standard Glossary of Terms used in Software Testing," *International Software Testing Qualifications Board*, vol. 3, no. 1, p. 30, September 1990.

[17] Haralambos Mouratidis, Paolo Giorgini and Gordon Manson, "Modelling Secure Multiagent Systems," in *Proceedings of the second international joint conference on autonomous agents and multiagent systems*, Melbourne, Australia, July 2003, pp. 859-866.

[18] Dr Hassan Shehata, Sabah Aly Darweesh, Abdallah Mahmoud, Amr M. Ismail, Eman G. El Twab, M. abbas aglan, M. Alaa Nour, M. Sameh, Ola Ashour, Reham M. Atef and Yousif W. Hanna, Software Agent for Programming language learning Advisory, *Ain Shams Universty, Faculty of Engineering, Department of Computer and System Engineering*. February 8, 2016, Diploma.