The Nucleus **54**, No. 2 (2017) 75-82

www.thenucleuspak.org.pk

The Nucleus
ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Methodology for Performance Evaluation of Distributed Multi Agent System

A. Ali[1,2], M. Aslam[1], J. I. Janjua[2] and M.U. Chaudhry[3]

[1]Department of Computer Science and Engineering, University of Engineering and Technology, Lahore, Pakistan

[2]Al-Khawarzimi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan

[3]Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea

aatka23@gmail.com; mtahir1713@gmail.com; jamshaid.janjua@kics.edu.pk; umarch@skku.edu

ARTICLE INFO

Article history :
Received : *18 October, 2016*
Accepted : *02 June, 2017*
Publsihed: *30 June, 2017*

Keywords :
Multi agent system
Performance evaluation
Distributed system

ABSTRACT

This paper presents a methodology for the assessment of the distributed multi agent system. It requires apt attention to check as if multi agent system is performing well. In recent research performance is taken as a rate, queuing system or some measure without capturing all characterization of the multi agent system, there is a lag for the proper evaluation of multi agent system. Our methodology takes the generic characteristics of the distributed multi agent system which can affect the overall efficiency of the system, and spotted some performance metrics. These metrics depict the clear image to measure the efficiency of the distributed multi agent systems. It is a step by step technique; in initial step, some performance metrics are derived from the generic properties of the multi agent system. In second step, the graphical interaction model of the real time multi agent system is developed. The Course Management System (CMS) and the Remote Patient Monitoring Systems (RPMS) are two distributed multi agent systems, their interaction models are generated using log files in Java Agent Development Environment. In the final step; the function from the graph theory is taken to validate the performance metrics for the both multi agent system CMS and RPMS.

1. Introduction

The flexible and adaptive multi agent systems are the most arduous systems today. They are demand of the today's business due to their magnificent properties. A multi-agent system is an association of synchronized, autonomous agents, which interact with each other in achieving common goals. In Multi-agent system (MAS), all agents communicate with each other by sending messages to each other in an expressive agent communication language. Agent communication language (ACL) [1] defines types of messages and their meaning that agent can exchange. Messages that agents communicate have semantic meaning which can be proposition, rules or action [2]. Differentiating from objects, agents just not only do message passing instead they are also capable of sharing complex information like ideas, rules, proposition etc. and also they have control over their internal state. The example of this can be a software agent in MAS, thatcan communicate with other agent without any plan of actionfrom environment and they must implement in order to collaborate [3]. Multi agent system based on distributed technology provides the intelligence to the distributed system, to make them more efficient and demanding. They form a rapidly change in field of computer science. To cope a distributed technology with multi agent system is of great importance. Distributed problem solving is in vogue now and replacing the parallel problem solving due to chattels of its reactivity, deliberation and hybridization [4].

Multi agent systems are covering a huge domain in business application and other fields too. Problems are also arising at continual bases as the technology flourishes. Likewise Communication, ontology problems in MAS, Performance evaluation also a critical issue yet to be address with all its directions. Analytical modeling using graph theory and the Petri net, simulation and the measuring techniques using GQM (Goal Question Metric) methods are the approaches in practice now a days. Measuring approaches have proposed such performance metrics that measure the only the communication and the organization properties of the MAS [5]. Performance is taken as rate or analyzing as a queuing system are the techniques where Erlang distribution is used to derive the mean response time of the system. System is taken as a Petri net to derive its efficiency [6]. istabt, isect functions are used to derive the connection cost metrics for the structural evaluation of the MASes [7], they actually measuring the reason of the communication drawback. Goal Question Metric technique needs to be defined independently for any specific implementation and the context or use [8].

*Corresponding author

The evaluation should be done with all its dimensions. As if the generic properties of the multi agent system are ignored, like cooperation, negotiation, autonomy, learning, intelligence, reactivity, social ability then the approach is lacking to evaluate the system overall. Analytical models [9] are addressing only the communication and organization of agents. A reliable methodology is essential that helps to evaluate MAS with all its generic characteristics that have impact on the performance. The innovation in the proposed model is the MPE (Multi Agent Performance Evaluation) model that measures all dominating properties of the multi agent system. Performance metrics are proposed to capture the whole characterization of the distributed multi agent system. We develop a methodology to test the input measure, output measure, process measure and the outcome measure of the distributed multi agent system. After that step, different experimentation is done to validate that performance metrics.

This paper is organized as follows; section 2 is the discussion of the related work for the performance evaluation of the MAS. Section 3 presents MPE model methodology with real time MAS test cases to validate the proposed performance metrics of generic characteristics as communication, autonomy, learning, and flexibility. Finally, conclusion and the future work are described.

2. Related Work

An aspect that should be taken into consideration is performance evaluation issues. The performance is actually the computational complexity of the underlying implementation of the MAS. It is the evaluation of some indicators; some basic indicators are computational time, throughput, concurrency, communication overhead, response time for a distributed system [10], they are also important in the distributed multi agent system. The researchers mostly check the functional properties of the multi-agent system and ignore the non-functional properties of the MAS, which collectively make the generics attributes of the multi agent distributed system [11]. Logically the performance issues depend upon directly the system s' unique underlying design and implementation technologies which also concerned to the non-functional properties of the MAS [12]. Researcher and engineers must show a lot of concerns to the performance issues as they are increasingly important for the multi-agent system design matters [13]. Multi agents are evaluated by the worth noting three methodologies, Analytical modeling, simulation, measurement.

In analytical modeling two techniques are basically used, Graphical models, Petri nets. Graphical models are analyzing the general organizational properties and the communication of the multi agent system. A model of three step process of observation, modeling and measure is adopted. A multi agent system for production planning and control in supply chain is focused to evaluate its organizational properties [14]. The observation model identifies when events in MAS are triggered, they are processed by the probe. The interactions between two agents are logged in the traces files, these files helps to draw an oriented graph of the system and calculate the metrics.

The graph is analyzed according to the organizational view of the multi agent system. It helps to understand the organizational structure of the multi agents system as distributed, centralized and mixed architectures. In this model the system dynamics and the proposed measures must be refocused for the evaluation betterment because the measure does not fulfill the generic characterization of the multi agent system [14].

In simulation model, the performance metrics are investigated and developed to evaluate the performance of the system [15]. Then the programming verification of the performance metrics is performed using two different environment configurations of the JAVA, Aglets and Jade. The system core technology is distributed to system and the architectures to verify are CORBA, DCOM and Java RMI. The implementations under analysis are Aglets, Jade as they provide a wide range of examples and quality of documentation. An idea similar to the Hamilton's Cycle is adopted with an exception that each time an agent receives the messages at the same time it sends a query to the agent broker to ask about the destination of the next message. Network configurations, agent combinations and number of messages sent simultaneously are the major complication in this scenario. 1024 agents placed on three hosts in different locations, have to communicate by exchanging 10 messages between each other simultaneously. The major issue is these concurrent messages are not fully maintained in this solution. The simulation is done through the variety of the experiments. These experiments easily verify the various connection metrics are appropriate to experiments in different configurations. The some parameters are used to derive the connection metrics for a distributed multi agent system. These are latency, stability, security and some other quantitative characteristics like network configuration. The latency, stability and security are considered as qualitative measures. Two methods are adopted here to derive the connection metrics [15].

A connection cost metric is derived in different ways. A system is considered to be a network having set of connected hosts denoted by H. Some functions are used to describe the qualitative measure of the system. Lt : H $\rightarrow[0,\infty)$, Istabt, isect : H $\rightarrow[0,1]$. This function respectively represents the average latency, degree of stability, degree of stability. Average latency function helps to minimize the host the time taken by the host to process received messages. To get the real time communication the small latency is required. The Istabt checks the degree of the stability of the system by having information about the host can do task successfully. The Isect function defines the probability that host is secure or not. These functions are dependent on the discrete time intervals of time. The aggregate of the all of these properties is Et : H x H $\rightarrow[0,\infty)$. This function is based on distance formula that gives the measurement of the distance between two hosts. The host distance measurement in a working network environment is done efficiently by calculating the proposed metrics in different experimental scenarios [15]. Network configuration is taken into account to build the different scenarios of the experimental setup. To derive the connection cost metrics an experiment setup has been made to derive the metrics from distributed system network configurations. Using this communication process there are, configuration like Single host, two local network hosts, two remote hosts, two remote hosts with firewall.

All of the agents and the Agent managers are placed on the single host. In the two host configuration case the number of the message exchanged are divided by two because of the two hosts. Two scenarios are followed in this configuration, in the best case all agents in the first case communicate with each other and then all agents on the second hosts. In the worst case, agents communicate one by one from two hosts though the network. Worst case is time consuming for the whole network. In the configuration of the two remote hosts with local broker, one host is made local to the agent manager. The agent manager shares the host with the agents. The broker is placed on the remote host. In the configuration of the “Two Local hosts with remote broker”, more than two hosts are used and the agent manager is placed on the third host. In the last case the “Host with local broker and the remote host”, the agent manager is local to the one of the two host used in the model. All these configurations are tested in two classes as connection without firewall and connection with firewall.

The generalized metrics and pre experiment suppositions are compared to the experimental results. Real system implementation behavior is predicted by using the combination of the low duty and the heavy duty results and the best and worst scenario metrics. Jade and Aglets verification results are used to simulate the performance of the model. Real-time message exchange is taken place to improve the calculations. The combination of the web service and the agent technology promises a better computing model [15].

In the Petri net model, the performance is taken as the mean response time of the system, does not address the properties of the MAS. There are many other accurate ways to calculate the transmission time of the system rather than the mention PERT based approach [16]. In the simulation method, the real world application evaluation turns to more complex while using that method, also the protocol used prevents the agents to cooperate more. The evaluation approach based on Petri nets is one of the mainly used approaches for the evaluation of the multi agent system. It is applied to a layered multi agent system and the layers are furthers associated with some other types of agents like manager, bidder and searchers. For communication between agents the time-out mechanisms are used. The method adopted here the approximation using the Erlang distribution which creates the number of distribution with different stages [16].

Measurement is an improvement tool for a particular project. Problem can be easily identified by applying the measurement to the process being improved as in Software process improvement (SPI), is one of the famous models to be used currently. When GQM method is applied the goals are refined to questions and then into metrics that adequately answer these questions. GQM is an efficient approach for assimilating goals of a specific project. It provides a measurement plan involving three steps [17]. Setting up goals for measurements, Goal oriented question asking, metrics should provide the best answer. All these techniques are worth noting and working proficiently, but the critical analysis tells some important aspects are neglecting while evaluating the system

In our approach we are creating an amalgam of the all three techniques, modeling, measurement and simulation where the modeling of the metrics using graphs theory measures. The generic attributes of the multi agent system are taken up to proposed the metrics as benchmark of the performance measurement. The maps for the measures are generated using real time actions of agents. The two test cases are developed using Java Agent Development Environment (JADE) [18], Remote Patient Monitoring System and Course Management system. The Proposed metrics are then applied to the test cases of MAS to simulate them. Our MPE model caters all techniques, analytical modeling, measurement and the simulation.

3. MAS Performance Evaluation Model: MPE Model

After having motivations and setting the goals in the previous study, now we propose our methodology for the performance evaluation of the distributed multi agent systems. Fig. 1 describes the steps involved in the performance evaluation method. The evaluation process is not the linear step; an intermediary step has been taken. That is first of all the measures for distributed MAS as a communication model are proposed and then the intelligence of the distributed MAS is measured into metrics.

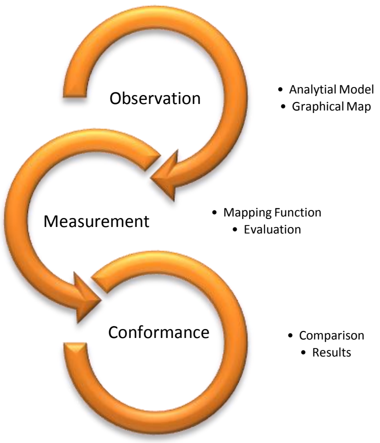

Fig. 1: MPE model: MAS performance evaluation model

In Fig. 1, the whole process of the proposed methodology is described the first phase shows the observation of the multi agent system; the related data is collected in this step. The generic attributes of the multi agent distributed system are observed to propose the metrics. The generic attributes like communication, autonomy, learning, and flexibility are particularly chosen that can address the all dimensions of the multi agent system. The second phase shows the measurement of the related data in form of the metrics and the third phase shows the conformance checking of the different measures of the MAS. The measurement is performed using the graph theory measures; metrics mapping functions are introduced to calculate the efficiency of the particular metric. After calculation of the metrics the best and worst case of the whole scenario are proposed along with the customization of any factor like autonomy, reactivity and the learning. MPE model can easily assess the all quality of the targeted system.

3.1 MAS Observation

Literature review showed evaluation of the multi agent systems leans to the design approaches and the development platforms. Our contribution is related to the general point of view of the multi agent system not concerned to its design or development.

The common properties of the multi agent systems are measured in our research works which influence its entire role. These properties are identified also in research [19], communication, autonomy, learning, and flexibility.

3.2 MAS Performance Measurements

Performance measurements are performed on the basis of the mapping function, as communication mapping function, autonomy mapping functions and the flexibility mapping functions. They take the generic attributes of the multi agent system. Using these functions the metrics for the attributes of multi agent system are derived. These metrics with their functions are given in Table 1.

Table 1: Performance metrics mapping functions

Multi agent Properties	Metrics	Metrics Mapping Functions
Communication	- no of Connections - Network Load - Network Complexity	- Indegree - OutDegree - Complexity Load - Connectivity
Autonomy	- Social Independence	- Order - HubDependence-Participation Coeficient
Learning	- Memorization, - Self-observation	- Self-Cycle - Cohesion index (c_i)
Flexibility	-persistent -communal	- Reachability - Clustering - Coeficient

3.2.1 Communication mapping functions

In graph theory, to measure the communication of the agent some metrics are proposed as a mapping function from the graph theory to complete the experimentation [20]. The communication metrics for a multi agent system to give an efficient performance are mentioned below:

3.2.1.1 No of connection

The count for the no of connection on an agent is placed to check how much connections are incoming and outgoing from an agent; we have two measures for this metric, In degree and out degree, they both reflect the input and claim of a particular agent to the communication act.

3.2.1.2 Network load

The network load is measured by the node function as per cycle in the network. In the graph theory [20] measures, it is named as the theta index in Eq.(1) below,

$$\theta = Q(G)/N \quad (1)$$

Q (G) is no of cycles per node; N is the total no of nodes in the network.

If $\theta > 1$, network load is greater

3.2.1.3 Network complexity

The network load of a multi agent system is measured by mapping the graph theory measure. It is the relationship between the number of connections and number of nodes (reflected as an agent) in a network of a multi agent system. This measure is named as a beta index in the graph theory.

$$\beta = V/N \quad (2)$$

Where V is no of connections and N is no of nodes in a network

If $\beta > 1$ network is more complex than other one

3.2.1.4 Net Communication

The net communication is the count for the total communication across a node in a network. According to graph theory [21], it is expressed by the relationship between the observed connections and the number of possible connections to a node. It is denoted as the Gamma Index, for planar graph in Eq. (3) below :

$$\gamma = e/3(v-2) \qquad (3)$$

Here e is the edge and the v is for the particular node.

For no planar graph, we have Eq. (4) mention below,

$$\gamma = e/ (v (v-2)/2) \qquad (4)$$

Here e is the edge and the v is for the particular node.

3.2.2 Autonomy Mapping Functions

3.2.2.1 Social Independence

The social independence is achieved in the agents having autonomy at high level. To check the social independence a nodal analysis is made for the particular node. The order of the node is the measure for this metric. The order of the node is measured by the no of incoming edges. Here is the measure for the nodal analysis :

Dependent nodes: have order that can be as low as 1

Independent node: order equal to the summation of the all orders of the other nodes in the graph

Isolate node: without connection/edges.

The order is measured in the graph theory [20] by the Eq. (4) given below,

$$ki = CD(i) = \sum Xij \qquad (5)$$

3.2.2.2 Memorization

The memorization and the self- observation are abilities of agent having 'Leaning' capability. An agent with memory can be a node of the oriented graph having self –loops. A node which starts and ends traversal on itself it has the ability of the memorization with having its on reference [20].

3.2.3 Flexibility Mapping Functions

3.2.3.1 Persistence

It is the ability of the particular node which is reachable throughout the entire network; it can be updated and queried about, so we can say it can maintain its state itself by having its reference .This ability is measured by reach ability of the node.

Reachability = % directly connected nodes in the entire graph, the nodes are directly connected when there exists a path between then, it means a sequence of adjacent vertices which starts with 'a' and ends with 'b'.

3.2.3.2 Communal

The communal metrics is measured by the clustering coefficient in the network. It is the probability of the interconnected adjacent nodes that expressed the tightly connected communities in the system. It is the ratio of the observed no of closed triplets to the number of possible triplets in the graph. According to graph theory in Eq. (6)

$$Ct=\lambda_G(V)/\tau_G(V) \qquad (6)$$

λ_G (V) = closed triplet

τ_G (V) = no of possible triplets

4. MAS Conformance

For the MAS conformance two test cases of the real time MAS are developed and then analytical models for those systems are generated and their performance is measured through the proposed performance metrics. The two MASes, Course Management System, Remote Patient Monitoring System are evaluated in the MPE model. Systems are developed on the JADE; their ontology's are developed according to the required scenarios. GUI is implemented by using JAVA Swing. A distributed JADE runtime is used for the action of the each agent in the above mentioned multi agent system. The action is actually the behaviour of a single agent. Ticker behaviour and a cyclic behaviour are adopted for the each agent. These MAS applications are more enhanced and can be exploiting as an industrial application. After having a real time application, an analytical model of the both systems is generated to measure the performance metrics. The performance metrics are then measured by the mapping functions of the graph theory. Some significant results are generated to conform the whole research methodology.

4.1 Mapping MAS properties to graph theory Measures

We have used the graph theory based performance measures to evaluate the performance of a real world distributed multi agent system. In the graph theory a system is taken as the graphical model. For the performance metrics, then proposed some mapping function from the graph theory measures and conform by using the real time application.

4.1.1 Mapping Course Management System

The system facilitate the teacher and the student both by using agent oriented technology, we have developed a real time multi agent system for course management system, In this real time application I have three agents, Teacher Agent, Intimate Agent, Student Agent. The teacher agent has the capability to put course for teacher with the constraint of the student Limit, Teacher will not able to offer the course if its registered student limit exceeds. When a teacher offer a course it will easily communicated to the student agent. The Student agent show his name and roll no by giving input of the course being registered and also check for if he had read its prerequisite course. If the student requests match the teacher offered subject then he will be able to register in the requesting subject. Fig. 2 shows the intimator agent took the student request and check if the any teacher offering the requested course from the student. If he finds the offered subject match to the student request then he will intimate the registration for the student particular subject to the student agent. Intimator is the control agent that will check the availability of the teacher or all student specifications.

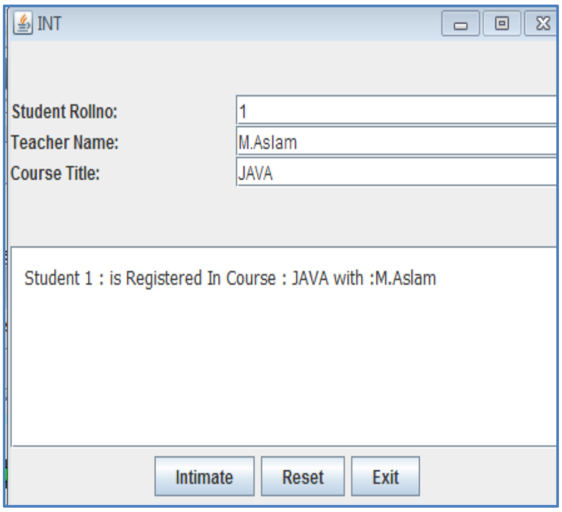

Fig. 2: Application for CMS: INT agent

Fig. 3 shows the Interaction model of the whole system is drawn using the graph Theory [20]. Four student agents labeled with ST1, ST2, ST3, ST4 are involved; they further interact with the intimator Agents INT1 and INT2. The intimator agents can directly interact with both Teacher agent labeled as T1, T2 and the student agent labeled as ST1, ST2, ST3, ST4.

Network Load is measured by the metric using Eq. (1), no of cycle as per Node /no of Nodes ⇒ 17/8=2.125. Network complexity is measured using Eq. (2) as no of Connectors/no of Nodes ⇒26/8 = 3.25.Autonomy is the mapping function for the no of the incoming edges in the graphical model. If the node is center node then it will have more interaction in form of incoming edges. The automated nodes or agent will project less no of edges to any centered node, they are socially independent. Fig. 4 shows graph for calculations for the autonomy in Course Management System.INT2 Agent has the autonomy more than other as it has more incoming edges than the other we can call it is the decision center of the system. So the system has only one decision center the level of the autonomy will be more better if it has more than one decision centers.

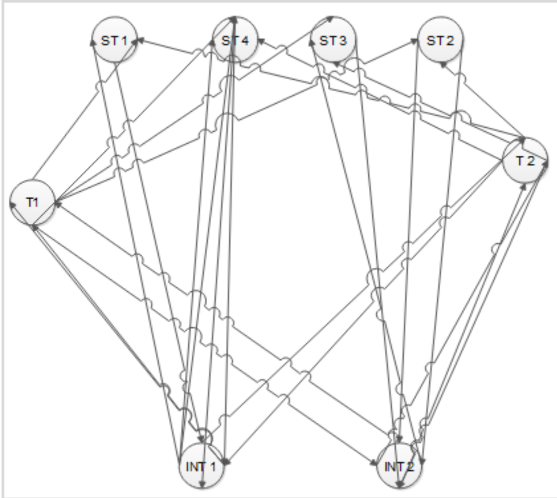

Fig. 3: Graphical model for CMS

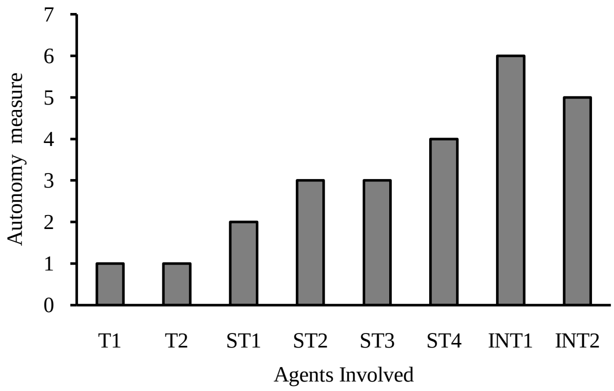

Fig. 4: Measuring autonomy in CMS

Learning is evaluated by checking the self-loop in whole system. Here the T1 is showing the greater capability of learning than other agents. It is capable of remembering its state. We find many self- loops here that are the successfully return to that node. Flexibility is the collective measure for the reactivity which is achieved through the persistence of the agent. Agent has the persistence show the reactivity to the environment. T and INT agent need resources to maintain its internal state, which is the reason for them being more reactive than other.

4.1.2 Mapping remote patient monitoring system

Multi agent system is developed to help in monitoring the patient remotely. Fig. 5 shows monitor agents continuously keep on sending the requests to check the patient condition remotely. If there comes some change beyond the normal measures for a patient the monitor agent will took the details of the patient, will invoke its friend agent and the relevant available doctor agent for the particular patient. The doctor agent will automatically prescribe the medicines for the remote agent or suggest any other advice for the patient.

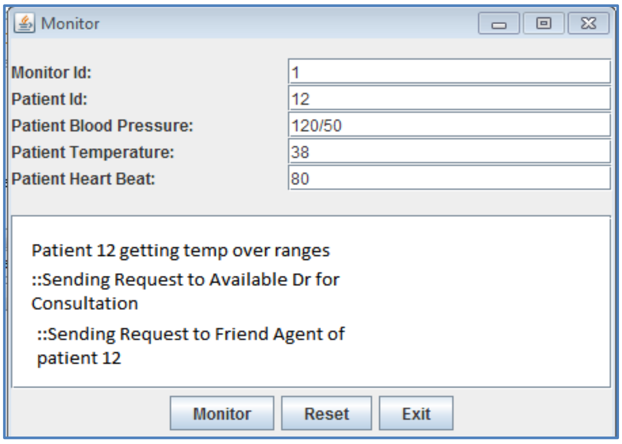

Fig. 5: Application for RPMS: Patient monitoring agent

Fig. 6 shows the interaction model of the RPMS is generated. Two Monitoring agents M1, M2 interacting with doctor agents dr1 and dr2, patient agents pt1 and pt2 and an Fr agent for the patient agent.

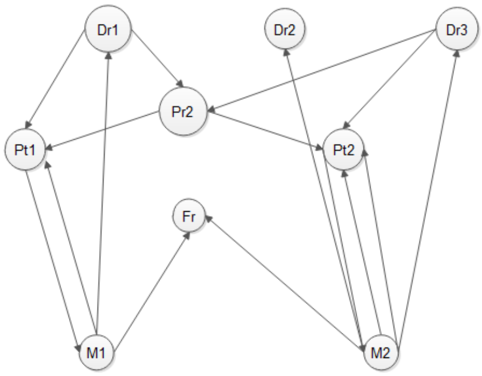

Fig. 6: Remote patient monitoring system model

Network Load of the RPMS is measured using Eq.(1), as no of Cycle per Node /no of Nodes ⇒10/8 = 1.25. Network Complexity of the RPMS is measured as no of Connectors/no of Nodes ⇒12/8= 1.5. Autonomy of the RPMS is calculated up to 4.

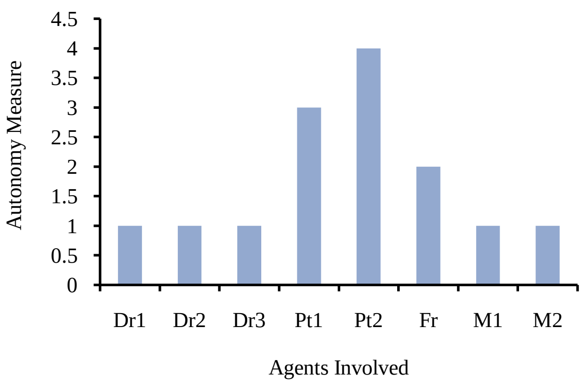

Fig. 7: Measuring autonomy for RPMS

PT2 is showing high levels of autonomy, which is monitor by the M1 and M2 agent. Learning is evaluated by checking the self-loop in whole system. The most agents in the system remember their state. Most of the agents in this system have the learning at the same level. They are performing well. Reactivity in the RPMS is also measured. The most reactive agent is the Fr agent, which must perceive its internal state.

5. Final Evaluation

We are considering the multi-agent system as an enterprise system, where an organization used to collect, manage and interpret data from many business activities. For evaluation we are proposing our best and worst cases by expending the performance measures proposed as the function of graphical models. By comparing the values of the real time applications of the distributed multi agent system and mapping the directed graphs of the multi agent system, we drawn some technical results. A CMS is the most precise automated and the reactive system, as we increase the reactivity in the RPMS system the automation slows down and the performance influenced. So we come up to some results, a reactive system is less automated and use more resources than a well performing automated system like CMS. So for a better performance we are moving to more automated and more efficient distributed multi agent system. Lesser the human intervention, lesser the resource utilization will show the better performing MAS.

6. Comparison

The existing practice to evaluate the performance evaluation of the multi agent system by creating the analytical model is just to address only the small scale organization by the multi agent system. The do not offer the powerful support for the evaluation of the generic attribute of the multi agent system that have the deep influential on the performance efficiency of the multi agent system .The distributed structures of the multi-agent system are highly complex structure, the existing model is just evaluating the only organization structure as it is distributed or centralized or mixed not evaluating its other performance measures. They are not mainly addressing the performance metrics of a distributed system. They are impartially analyzing the organizational structure to evaluate a performance of the multi agent system. The metrics does not capture the all characterization of the multi agent system. In our MPE model, we have generically created the analytical measurement model of our real time distributed multi agent system, assess their performance metrics that highly influence their performance and then evaluate it by generating the mapping through the graph theory measure [20]. The performance metrics are then calculated to give the final result of the multi agent system, which we are going to evaluate. The multi agent evaluation cases are given in

Table 2 that are taken as point of reference for performance evaluation.

Characterization is measured more than one metrics given in Table 2. Utmost, it is an adequate and generic model for evaluating the performance evaluation of the distributed multi agent system.

Table 2: MPE evaluation cases

Evaluation Parameters	Best Case	Worst Case
Network Complexity	low	high
Autonomy	high	low
Learning	high	low
Flexibility	Under demand	low

Table 3: MPE final results

Evaluation parameters	CMS	RPMS
Network Complexity	3.25	1.5
Network Load	2.125	1.25
Autonomy	6	4
Learning	4	2
Persistence/Reactivity	6	8

7. Conclusion

Distributed multi agent systems are highly complicated structures, coming in the technology trends. The performance metrics for the distributed multi agent system are proposed. We have evaluated our real time distributed multi agent system as a test application, by using our analytical model. For setting standards in the social networks we have also include its performance metric communal, which will be analyzed in our future work. Also we want to enhance our technique for the other architecture which is coming in vogue now a day.

References

[1] G. A. Kaminka, "Robots are Agents, Too!", AgentLink News: 16–17, December, 2004.

[2] Y. Kubera, P. Mathieu and S. Picault, "Everything can be Agent!," in Proceedings of the ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2010) , Toronto, Canada, International Foundation for Autonomous Agents and Multiagent Systems, vol. 1, pp. 1547-1548, 2010.

[3] S. J. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach", 2nd Ed., Pearson Education, 2003.

[4] Coulouris, George F., Jean Dollimore, and Tim Kindberg, “Distributed systems: Concepts and design”, Pearson Education, 2005.

[5] F.B. Hmida, W.L. Chaari, "Aspect-Based Multiagent Systems Observation for Performance Evaluation", Multi Conference on Computer Science and Information Systems, Proc. of Intelligent Systems and Agents, July 22-24, 2008, Amsterdam, Netherlands, International Association for Development of the Information Society, IADIS, pp. 172-176, 2008.

[6] T. Babczyński and J. Magott, "Estimation of mean response time of multi-agent systems", International Federation for Information Processing, Software Engineering Techniques: Design for Quality, Boston: Springer, US, vol. 227, pp. 109-113, 2006.

[7] D. Król and M. Zelmozer, “A comparison of performance-evaluating strategies for data exchange in multi-agent system”, KES-AMSTA: Proc. of the 2nd KES Int. Conf. on Agent and Multi-agent Systems: Technologies and Applications, March 26-28, 2008, Incheon, Korea, Springer-Verlag Berlin, Heidelberg, L NAI, vol. 4953, pp. 793-802, 2008.

[8] P.D. Bitonto and M. Laterza, “An evaluation method for multi-agent systems”, KES-AMSTA: Proc. of the 4th KES Int. Symp. on Agent and Multi-agent Systems: Technologies and Applications, June 23-25, 2010, Gdynia, Poland, Springer-Berlin, Heidelberg, LNCS, vol. 6070, pp. 32-41, 2010.

[9] F.B. Hmida and W.L. Chaari, “Evaluation of Communication in Multiagent Systems for Supply Chain Planning and Control”, 26th IEEE International Conference on Tools with Artifical Intelligence, Limassol, Cyprus, pp. 408–412, November 10-12, 2014.

[10] P. Bukh and R. Jain, “The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and mode*ling*, New York: Wiley-Interscience, April 1991.

[11] L.C. Lee and H.S. Nwana, "The stability, scalability and performance of multi-agent systems", BT Technology Journal, vol. 16, no. 3, pp. 94-103, July 1998.

[12] P.J. Turner and N.R. Jennings, “Improving the Scalability of Multi-agent Systems", Infrastructure for agents, multi-agent systems and scalable multi-agent systems, AGENTS, 2000, UK, Springer-Berlin, Heidelberg, LNCS, vol. 1887, pp. 246-262, 2000.

[13] M. Wooldridge, “An Introduction to MultiAgent Systems”, 2nd Ed., John Wiley & Sons, 2009.

[14] F.B. Hmida and W.L. Chaari, "Graph-based evaluation of organization in multi agent systems", 9th Int. Conf. of Modeling, Optimization and Simulation, MOSIM’12, Bordeaux, France, June 6-8, 2012.

[15] D. Król and M. M. Zelmozer, "Structural Performance Evaluation of Multi-Agent Systems", JUCS, vol. 14, no. 7, pp. 1154-1178, 2008.

[16] T. Babczyński and J. Magott, "PERT based approach to performance analysis of multi–agent systems", Int. Conf. on Artificial Intelligence and Soft Computing, ICAISC, 2006, Springer-Berlin, Heidelberg, LNCS, vol. 4029, pp. 1040-1049, 2006.

[17] T. Punter, "Software Measurement with GQM in Small and Medium Enterprise," Software-Metriken in der Praxis-Tagungsband der DASMA-KonferenzMetrikon , Shaker Verlag, Aachen, 2001.

[18] F Bellifemine and G Caire, "JADE: A software framework for developing multi-agent applications. Lessons learned”, Information and Software Technology, vol. 50, no. 1-2, pp. 10-21, January, 2008.

[19] K. Potiron and P Tailliber, “From fault classification to fault tolerance for multi-agent systems”, Springer, vol. 8, pp.19, 2013.

[20] F. Harary, “Graph Theory”, Reading, MA: Addison-Wesley, pp. 17-18, 1994.