

Measures for Testing the Reactivity Property of a Software Agent

N.Sivakumar

Department of Computer Science and Engineering
Pondicherry Engineering College
Puducherry, INDIA.

K.Vivekanandan

Department of Computer Science and Engineering
Pondicherry Engineering College
Puducherry, INDIA.

Abstract—Agent technology is meant for developing complex distributed applications. Software agents are the key building blocks of a Multi-Agent System (MAS). Software agents are unique in its nature as it possesses certain distinctive properties such as Pro-activity, Reactivity, Social-ability, Mobility etc., Agent's behavior might differ for same input at different cases and thus testing an agent and to evaluate the quality of an agent is a tedious task. Thus the measures to evaluate the quality characteristics of an agent and to evaluate the agent behavior are lacking. The main objective of the paper is to come out with a set of measures to evaluate agent's characteristics in particular the reactive property, so that the quality of an agent can be determined.

Keywords-Software Agent; Multi-agent system; Software Testing.

I. INTRODUCTION

Agent technology is one of the rapidly growing fields of information technology and possesses huge scope for research both in industry as well as in academic level. Software agents can be simply defined as an abstraction to describe computer programs that acts on behalf of another program or user either directly or indirectly [1]. Software agent is endowed with intelligence in such a way that it adapts and learns in order to solve complex problems and to achieve their goals. Software agents are widely employed to greater extent for the realization of various complex application systems such as Electronic commerce, Information retrieval and Virtual corporations. For example in an online shopping system the software agent help the internet users to find services that are related to the one they just used. Though agent oriented systems has progressive growth, there is a lack in its uptake as there is no proper testing mechanism for testing an agent based system [2].

Software quality can be examined in different perspective such as conformance to customers' requirements and development process quality such as requirement, design, implementation, test and maintenance quality [3].The metrics are the quantitative measures for the evaluation of a software quality attributes. Applying metrics [4] [5] for a software agent is a complex task as every agent exhibit cognitive characteristics such as autonomy, reactivity, pro-activeness, social-ability etc.

- Autonomy – Self-control over actions and states.
- Reactivity – Responsiveness to changes in environment

- Pro-activity – Exhibit goal-oriented behavior
- Social ability – Collaboration leading to goal achievement.

Software quality of an agent-based system can neither be easily measured, nor clearly defined. Measuring software quality of an agent depends upon the ability to describe the agent characteristics such as autonomy, reactivity, pro-activeness and collaboration. A set of measures for evaluating the software agent's autonomy [6] [9], pro-activity [7], social-ability[8] [9], has been dealt in the literature. In this paper, a set of measures for evaluating the software agent's reactivity property, considering its associated attributes has been proposed.

II. RELATED WORK

A. Software Agent and its Properties[1]

Software agent is an autonomous entity driven by beliefs, goals, capabilities and plans. An agent has a number of agency properties such as autonomy, pro-activity, reactivity, social-ability, learnability, mobility.

Autonomous- Agents should operate without the intervention of external elements (other agents or humans). Agents have their control over their actions and internal states.

Proactivity - Agents should exhibit goal directed behavior such that their performed actions cause beneficial changes to the environment. This capability often requires the agent to anticipate future situations (e.g. using prediction) rather than just simply responding to changes within their environment.

Reactivity - Agents perceive their environment and respond in a timely fashion to changes that may occur.

Social Ability- A software agent is able to use communication as a basis to signal interest or information to either homogeneous or heterogeneous agents that constitute a part of its environment. The agent may work towards a single global goal or separate individual goals.

Mobility – The ability of being able to migrate in a self-directed way from one host platform to another

B. Quality of Software Agent[2][3][4]

In general, the quality of the software depends on the functional and non-functional metrics. Measuring quality is a tedious and also important task of software project

management. When it comes to Multi-Agent System (MAS), the quality is majorly based on how the agents involved in the system works as a separate entity and also in co-ordination with other agents.

To test the functionality of an agent, it is very important to evaluate the characteristics of an agent such as autonomy, pro-activity, reactivity and social-ability [6]. But evaluating the agent characteristics is not a simple task because an agent reacts differently for the same input in different scenario.

C. Measuring Autonomy of an agent[7][10]

Agent autonomy is a characteristic that is interpreted as freedom from external intervention, oversight, or control. Autonomous agents are agents that are able to work on behalf of their user without the need for any external guidance. Agent autonomy considers three important attributes such as self-control, functional dependence and evolution capability.

1) Self-control

Self-control ability is identified by the level of control that the agent has over its own state and behavior. Self-control attributes can be measured using the following measures

- Structural Complexity
- Internal State Size
- Behavior Complexity

2) Functional dependence

Functional dependence is related to executive tasks requiring an action that the agent has to perform on behalf of either the user it represents or other agents. Functional dependence attributes can be measured using the following measures

- Executive Message Ratio

3) Evolution capability

Evolution capability of an agent refers to the capability of the agent to adapt to meet new requirements and to take necessary actions to self-adjust to new goals. Evolution capability attributes can be measured using the following measures

- State Update Capacity
- Frequency of state Update

D. Measuring Pro-activity of an agent[8]

Agent pro-activity considers three important attributes such as initiative, interaction and reaction.

1) Initiative

Initiative is the agent's ability to take an action with the aim of achieving its goal. Initiatives can be measured using the following measures

- Number of Roles
- Number of Goals
- Messages to achieve the goals

2) Interaction

Interaction is the agent's ability to interact with other agents, the user and its environment. Interaction can be measured using the following measures

- Method per Class
- Number of Message Type

3) Reaction

Reaction is the ability to react to a stimulus from the environment, according to stimulus/response behavior. Reaction can be measured using the following measures

- Number of Processed Requests
- Agent Operations Complexity

E. Measuring Social-ability of an agent[9][10]

An agent's social ability is represented by the attributes related to communication, cooperation and negotiation.

1) Communication

The ability of communication is identified by the reception and delivery of messages by the agent to achieve its goals. Communication can be measured using the following measures

- Response for Message
- Average Message Size
- Incoming Message
- Outgoing Message

2) Cooperation

Cooperation indicates the agent's ability to respond to the services requested by other agents and to offer services to other agents. Cooperation can be measured using the following measures

- Services Requests Rejected by the Agent
- Agent Services Advertised

3) Negotiation

Negotiation is the agent's ability to make commitments, resolve conflicts and reach agreements with other agents to assure the accomplishment of its goals. Negotiation can be measured using the following measures

- Agent Goals Achievement
- Messages by a Requested Service
- Messages Sent to Request a Service

III. PROPOSED WORK

Software quality is an important non-functional requirement for any software and agent-based software is not an exception. Software quality of an agent-based system is depends on the characteristics of an agent such as autonomy, pro-activity, reactivity, social ability, intelligence.

Although there are various measures for evaluating agent autonomy and social ability, a comprehensive set of measures has not yet been developed for measuring the reactivity of an agent. Reactivity of a software agent is defined as the ability to perceive its environment and respond in a timely fashion to any

environmental changes. The main objective of the proposed work is to present a set of measures for evaluating the reactivity characteristic of an agent which cannot be measured using a single metric but at different levels [11] such as

- Interaction level
- Communication level
- Perception level

A. Interaction Level

Interaction level expresses the activity of agents during their interaction. It directly reflects the measure of reactivity because when agents interact with each other, the reactivity of agents depends on each other's interaction level. Under different situation, agents might react differently with other agents and their environment. A high interaction level might indicate that the agent is able to react to multiple situations. The metric suit for interaction level consists of,

- Methods per Class (MC)
- Number of Message Types (NMT)

1) Methods per Class (MC)

MC measures the number of methods implemented within the agent enabling it to achieve its goals. If the agent has many different methods for achieving a goal, it will be able to interact better and will have a better chance of react to achieve its goals. The method per class is calculated at the method level and calculated using the parameters such as, the number of conditional statements, the number of loop statements, local and global variables, read and write variables. The average of all the parameters mentioned will give us the value of the Method per class metric.

2) Number of Message Type (NMT)

This metric measured the number of different type of agent messages that can be resolved or catered by the agent. The more message types an agent could handle, the better it has developed its interaction capability and increases the reactivity of agents. The total number of messages is given by the formula, $NMT = IM + OM$, where IM and OM is the number of unique incoming and outgoing message type respectively and it is calculated at the class level.

B. Communication level

The level of conversation may view as the amount of messages that have to be transferred to and from, in order to maintain a meaningful communication link or accomplish some objectives. High communication intensity can affect the reactivity of an agent as it may means that the agent has spent much of its resources in the handling of incoming request from other agents for its service thus making it harder to modify. It could also means the agent has much outgoing request to other agents for their services, indicating an excessive coupling design. Agents should have minimal communication as most agents will only interact with the service providing agents and when providing services or detecting and responding to the environment changes. Agents usually communicate with the services yellow page to search for required service and thus do not required to send messages to all other agents in the system

for services. The following are the agent communication level metrics,

- Response For Message (RFM)
- Incoming Message (IM)
- Outgoing Message (OM)

1) Response for Message (RFM)

RFM measures the amount of messages that are invoked in response to a message received by the agent. To process the incoming message, new messages might be sent to another agent requesting new services. It is calculated at the method level and it is calculated using the parameters such as the external calls and the internal calls. Response for message is the average of the total number of the external calls and the total number of the internal calls.

2) Incoming Message (IM)

IM measures the relation of incoming messages to agent communication during its lifetime. Higher values indicate that the agent has more dependent agents requiring its services. This measure is calculated at the class level.

3) Outgoing Message (OM)

OM measures the relationship between direct outgoing messages and agent communication during its lifetime. Higher values could indicate that the agent is dependent on other agents. This measure is calculated at the class level.

C. Perception level

The level of understanding the environment is termed as Perception. Perception directly or indirectly influences the intelligence of agents. The agents should be updated with the events occurring in the environment. Higher level of perception ratio indicates that the agent is more reactive because the agent gets all the information to itself. So that the messages sent to other agents for requesting the services gets reduced. This implies that the agent is more reactive. The metric suit for perception level consists of,

- Knowledge Usage (KUG)
- Knowledge Update (KUP)

1) Knowledge Usage (KUG)

Knowledge usage measures the average number of internal agent attributes used in the decision statements inside the agent methods. It is dependent on the parameters such as the read variables, read methods. Variables which affect more decision making process would have a stronger influence over the agent behavior. Given more of the decision making process uses the internal states, then the agent is said to be greater affected by the perception level and might be less predictable if the values changed frequently. Higher values indicate that the agent system is more complex, thus agents react with each other performing many services.

2) Knowledge Update (KUP)

Derive from live variables, this metric count the number of statement that will update the variables in the agent. Each variable is dependent on different event occurrence, where the

event would change the variable value, thus agent internal states.

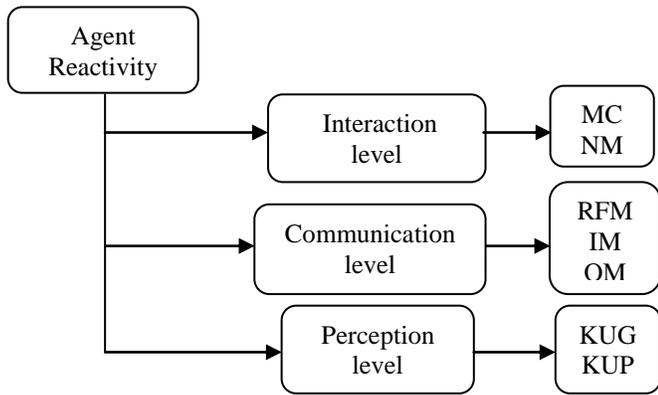


Figure 1. Agent Reactivity Levels with Metrics

IV. IMPLEMENTATION

Quality of an agent-based system is based on how agent adopts its properties such as autonomy, pro-activity, reactivity, social-ability, learnability. A tool that calculates the attributes of agent reactivity property at various levels such as Interaction, Perception and Communication level has been implemented.

The implementation focuses on developing agent reactivity calculator tool that determines and collects agent specific metric data according to above mentioned levels. The tool is designed to evaluate metrics that relate to quality of the agent oriented programs in particular the reactivity property. The calculated metric values are stored in a database for further reference and analysis. Javais used as a front-end tool to provide a user-friendly, interactive interface.

The agent based projects to be analyzed have been developed using JADE [12] framework and FIPA standards. These projects shouldn't have any syntax errors and the code should be capable of being executed independently.

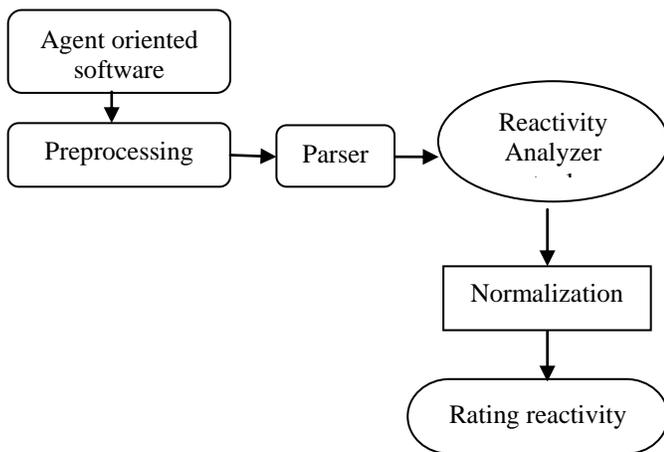


Figure 2. System Design

1) Agent Oriented Software

The input to the system is the agent based system which has to be analyzed and they have been developed using JADE framework and FIPA standards. These systems shouldn't have any syntax errors and the code should be capable of being executed independently.

2) Preprocessing

A preprocessor is designed to remove all spaces and statements that would not be useful for the purpose of metrics calculation. The result from this preprocessor is then sent to a parser

3) Parser

The functions of the parser are to construct the Abstract Syntax Tree which is required for the metric calculation. The ANTLR (Another Tool for Language Recognition) framework generates the necessary java class files. The parser recognizes the language and creates the tree. The tokens present in the tree are also separated based on their types.

4) Agent Reactivity Analyzer

The Agent reactivity analyzer tool is designed to evaluate metrics that relate to reactivity of the agent oriented programs at various levels such as Interaction level, Perception level, Communication level and Reaction level. The calculated metric values are stored in a database for further reference and analysis.

5) Normalizing the Results

To measure the quality, the measured metrics value will be expressed in the range of 0 and 1 (where 0 means a poor result for the measure and 1 means a good result). The process of transforming our index from its value into a range of 0 and 1 is called normalization. The calculated metrics at each level is normalized in the range of 0 and 1 using the following formula $N=d/\sqrt{d^2+a}$, where 'd' is the similarity between index and 'a' is the actual value. The values obtained after normalization can be rated using the tabulation given below.

6) Rating Reactivity

After obtaining the actual values of all the metrics proposed, they should be rated. If the value interval ranges from 0.00 – 0.20, 0.20 – 0.40, 0.40 – 0.60, 0.60 – 0.80, 0.80 – 1.00, it is tagged as Very less Reactive (VLR), Less Reactive (LR), Average Reactive (AR), High Reactive (HR), and Very High Reactive (VHR) respectively. The following tabular column shows the value ranges.

TABLE I. RATING REACTIVITY

Value internal	Rating	Acronym
0.00 – 0.20	Very Less Reactive	VLR
0.20 – 0.40	Less Reactive	LR
0.40 – 0.60	Average Reactive	AR
0.60 – 0.80	High Reactive	HR
0.80 – 1.00	Very High Reactive	VHR

V. CASE STUDY

Agent-based Online shopping system involving five types of agents such as interface agent, buyer agent, expert agent,

evaluation agent and collaboration agent is developed. The overall goal of the system is to analyze a customer's current requirements and to find the most suitable commodity for him/her. These agents collaborate with each other by message delivery mechanism and make the whole system works together. The detailed functions of each agent in the shopping system are described as follows.

1) *Interface Agent(A1)*

The main work of the interface agent is bidirectional communication between the shopping system and customers. In order to collect and analyse the customer's current needs, the interface agent asks him/her some specially designed questions about the commodities. In the shopping system, assuming that the customer does not have enough domain knowledge to answer quantitative questions regarding the technical details about the commodity, the system has to inquire some qualitative ones instead. For example, the system will ask the customer to express his need on the display feature.

2) *Buyer Agent(A2)*

Buyer agent is a mobile agent, which can migrate to the electronic marketplace and search for the commodity information from multiple sellers. When it searches out one seller, it will ask for offers about the commodity from the respective seller. After the buyer agent gets all offers, it will return back and store the commodity information in the internal commodity database.

3) *Expert Agent(A3)*

The expert agent provides the communication interface with human experts, by which the experts can embed their personal knowledge into the system and give a score of a commodity in each qualitative need defined before. With the expert agent, the system can collect opinions from different experts to give more objective suggestions. Then the expert agent will convert them into a specially designed internal form for knowledge representation. However, human experts seldom reach exactly the same conclusions. They may give different scores of the same commodity in the same qualitative need since their preferences are different. In order to resolve this problem, the system synthesizes all the expert's opinions and assigns the same weights for them in the system implementation. In this way, the expert agent can transfer each commodity to a rank form and calculate its optimality accordingly.

4) *Evaluation Agent(A4)*

After receiving the offers of all commodities from the sellers, the evaluation agent will have comparison mechanism to evaluate each commodity in order to make the best possible selection of all the supplied commodities. Since shopping is not just searching for a lower price commodity. There is something else that should be taken into considerations like quality, reliability, brand, service, etc. Based on the multi-attribute evaluation model, the evaluation agent calculates the utility value of each commodity and selects one that has maximal utility value as the recommended commodity.

5) *Collaboration Agent(A5)*

User-system interaction is an important factor in achieving optimal recommendation. During the interaction, the consumer

can give more feedback to the system by updating his/her current needs until the consumer is satisfied with the shopping result. However, the frequent user-system interactions inevitably take time. In the system, collaboration agent is designed to reduce the time of user-system interaction. The collaboration agent is based on the consumer-based collaboration approach which first compares the need pattern of the current customer to the ones previously recorded and then system recommends the commodities selected by the similar consumers to the current customer.

VI. RESULT INTERPRETATION

Reaction is the ability to react to an action from the environment according to the action behavior. Agents react appropriately according to the context in which they operate. The agent-based online shopping system involving five agents such as Interface agent, Buyer agent, Expert agent, Evaluation agent and Collaboration agent has been taken as a case study to evaluate the reactivity property. Agent-based online shopping system is given as an input to the reactivity analyzer tool (ref Figure. 4).

The tool starts with preprocessing the agent code and parses it as required to calculate the reactivity. Every agent involved in online shopping system such as Interface agent (A1), Buyer agent (A2), Expert agent (A3), Evaluation agent (A4) and Collaboration agent (A5) are evaluated with the metrics related to various levels such as Interaction level, Communication level, Perception level and Reaction level. The metric value of the measures at various levels for all the five agents are tabulated in Table II.

The metrics value in Table II is normalized in such a way that the values are expressed in the range of 0 and 1 (where 0 means a poor result for the measure and 1 means a good result). For example, in the interaction level, if the normalized value is in the range of 0.00 to 0.20 then, the interpretation is, the agent is very less interactive among other agents. Similarly if the normalized value is in the range of 0.80 to 1.00 then, the interpretation is, the agent is very high interactive among other agents. The complete range of possible normalized values and their respective rating is tabulated in Table III. The normalized value of the metrics calculated and their corresponding ratings are tabulated in Table IV. From Table IV, we interpret that agent A2 i.e. Buyer agent is very high interactive, very high communicative, very high perceptive. Thus considering all levels we understood that buyer agent is more reactive towards the environment and behaves in a timely fashion. Similarly all the agents involved and their corresponding reactivity rating is tabulated in Table IV.

The comparative analysis of various agents and their corresponding evaluation measures at various levels such as Interaction level, Communication level and Perception level are represented by the chart in figure 3, figure 4 and figure 5 respectively. The overall Reactivity rating is represented in figure 6. From figure.6 we interpret that every agent in the online shopping system are reactive in nature whereas the buyer agent (A2) is more reactive than any other agents as the agent involves more negotiation and co-ordination with other agents.

TABLE II. METRIC VALUES AT VARIOUS LEVEL

Agent	Interaction level		Communication level			Perception level	
	MC	NMT	RFM	IM	OM	KUG	KUP
A1	0.4	4.0	1.0	3.0	3.8	1.1	4.3
A2	0.7	6.0	0.9	1.8	1.8	1.2	4.5
A3	0.4	4.3	1.0	2.0	2.0	1.1	4.1
A4	0.5	4.5	0.8	1.8	1.7	1.2	4.5
A5	0.6	5.5	0.9	1.8	1.8	1.2	4.5

TABLE III. METRIC RATING VALUES

Value range	0.00 – 0.20	0.20 – 0.40	0.40 – 0.60	0.60 – 0.80	0.80 – 1.00
Interaction level	Very less Interaction (VLI)	Less Interaction (LI)	Average Interaction (AI)	High Interaction (HI)	Very high Interaction (VHI)
Perception level	Very less Perception (VLP)	Less Perception (LP)	Average Perception (AP)	High Perception (HP)	Very high Perception (VHP)
Communication level	Very less Communication (VLC)	Less Communication (LC)	Average Communication (AC)	High Communication (HC)	Very high Communication (VHC)
Reactivity	Very less Reactive (VLR)	Less Reactive (LR)	Average Reactive (AR)	High Reactive (HR)	Very high Reactive (VHR)

TABLE IV. NORMALIZED VALUES AT EACH LEVEL

Agent	Interaction level		Communication level		Perception level		Overall Reactivity
	Normalized interaction values	Rating	Normalized Communication values	Rating	Normalized Perception values	Rating	
A1	0.64	HI	1.00	VHC	0.99	VHP	0.87 (VHR)
A2	0.90	VHI	1.00	VHC	1.00	VHP	0.96 (VHR)
A3	0.72	HI	1.00	VHC	0.91	VHP	0.87 (VHR)
A4	0.76	HI	0.96	VHC	1.00	VHP	0.89 (VHR)
A5	0.76	HI	0.99	VHC	0.99	VHP	0.81 (VHR)

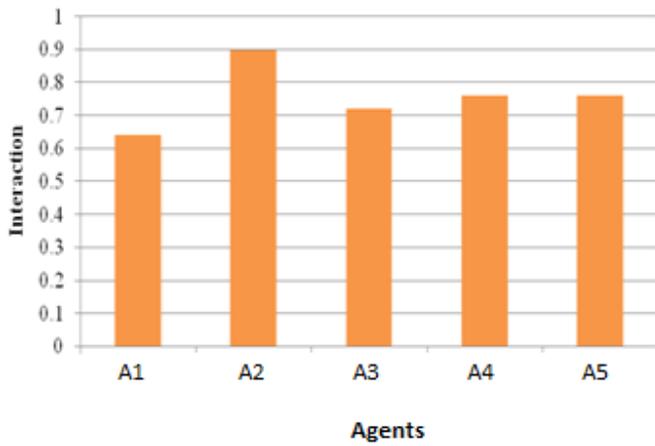


Figure 3. Interaction Values for Various Agents

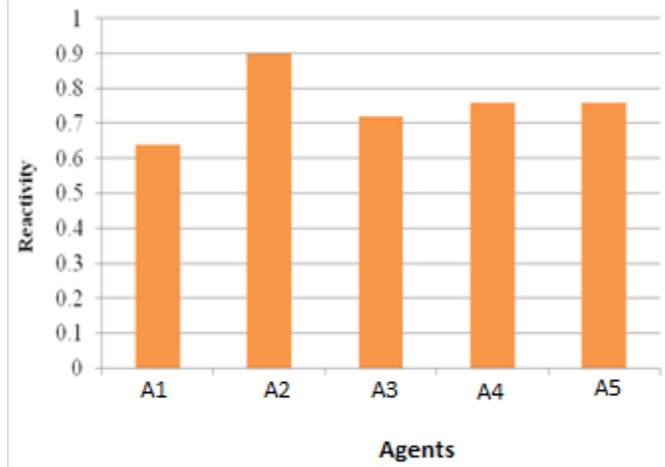


Figure 6. Overall Reactivity Values for Various Agents

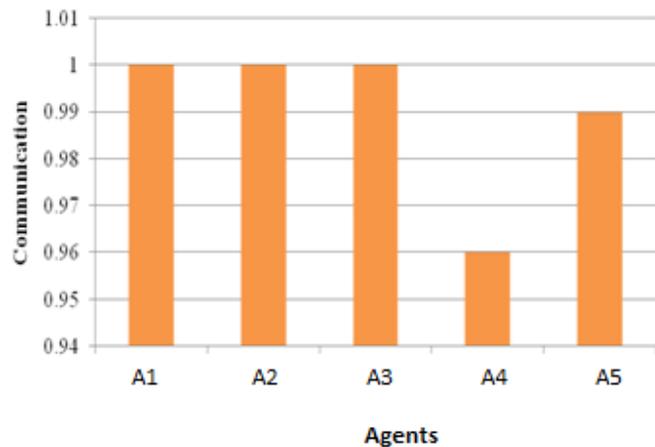


Figure 4. Communication Values for Various Agents



Figure 5. Perception Values for Various Agents

VII. CONCLUSION

The successfulness of any software is acknowledged based on its quality. Determining the quality of a software is not a simple task and it can be achieved only with suitable metrics. Since the quality of an Multi-Agent System is dependent on how the agents involved in the system works, it is the prime importance to analyse the properties of agent such as autonomy, pro-activity, reactivity and social-ability. From the literature it is understood that the various measures for evaluating autonomy, pro-activity and social-ability has already been proposed and thereby the need for metrics for evaluating reactivity property is implicitly known. In this paper, a thorough study on agent based system and the role of agent characteristics in particular the reactivity property in evaluating the quality measure is made. The set of measures for evaluating the reactivity property, considering its associated attributes at various levels such as interaction, communication and perception level is identified and implemented. An online shopping system involving five agents has been taken as case study to evaluate the set of measures identified for measuring the reactivity property and the results are encouraging.

REFERENCES

- [1] Nwana.G, "Software Agents: An Overview", The Knowledge Engineering Review, 11(3), pages 205-244.
- [2] I. Duncan, and T. Storer, "Agent testing in an ambient world", in T. Strang, V. Cahill, and A. Quigley (eds.), Pervasive 2006 Workshop Proceedings, Dublin, Eire, May 2006, pp. 757764.
- [3] R. Dumke, R. Koeppe, and C. Wille, "Software Agent Measurement and Self-Measuring Agent-Based Systems," Preprint No 11. Fakultät für Informatik, Otto-von-Guericke-Universität, Magdeburg (2000).
- [4] J. D.Cooper and M. J. Fisher, (eds.) "Software Quality Management", Petrocell Books, New York (1979), pp. 127-142.
- [5] B. Far, and T. Wanyama, "Metrics For Agent-Based Software Development", Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2003), May, 2003, pp. 1297-1300.
- [6] D. Franklin, and A. Abrao, "Measuring Software Agent's Intelligence", Proc. International Conference: Advances in Infrastructure for Electronical Business, Science and Education on the Internet, L'Aquila, Italy, August, 2000.

- [7] F. Alonso, J. L. Fuertes, L. Martínez, and H. Soza, "Towards a Set of Measures for Evaluating Software Agent Autonomy," Proc. of the 7th Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering, Springer, 2010.
- [8] Alonso, J. L. Fuertes, L. Martínez, and H. Soza, "Measuring the Proactivity of Software Agent" Proc. of the 5th International conference on Software engineering Advances, IEEE, 2010
- [9] F. Alonso, J. L. Fuertes, L. Martínez, and H. Soza, "Measuring the Social Ability of Software Agents," Proc. of the Sixth International Conference on Software Engineering Research, Management and Applications, Prague, Czech Republic (2008), pp. 3–10.
- [10] Fernando Alonso, Jose L. Fuertes, Loic Martinex and Hector Soza, "Evaluating Software Agent Quality: Measuring Social Ability and Autonomy", Innovations in Computing Sciences and Software Engineering, Springer, 2010.
- [11] K. Shin, "Software Agents Metrics. A Preliminary Study & Development of a Metric Analyzer," Project Report No. H98010. Dept. Computer Science, School of Computing, National University of Singapore (2003/2004).
- [12] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, "Developing Multiagent Systems with JADE", John Wiley & Sons, Inc., 2007.