A REFINED METRIC SUITE FOR A MULTI AGENT SYSTEM

Antoinette[1], Archana[2], Mathiyarasi[3], Dhanavanthini[4]

[1]*Information Technology, Pondicherry University*
[2]*Information Technology, Pondicherry University*
[3]*Information Technology, Pondicherry University*
[4]*Information Technology, Pondicherry University*

Abstract

Metrics are the basic factor for the evaluation process of an agent software .The evaluation process are complex and the available metrics for measuring the agent characteristic are in sufficient. This is due to the factor that the agents are unpredictable in a multi-agent system (MAS).In this paper we have done a detailed study about the agent-oriented methodologies and agent-oriented metrics in a suitable environment.

Keywords: *agent metrics; MAS; AOSE;*

---***---

1. INTRODUCTION

Artificial Intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs [2]. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable. Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people and some machines. Artificial Intelligence helps the machines to find solutions to complex problems in a more human-like fashion.

Artificial Intelligence ability is to interact with the real world is to perceive, understand, and act. Agent is an autonomous entity which observes through sensors and acts upon an environment using actuators. A language is usually considered object-based if it includes the basic capabilities for an object: identity, properties, and attributes [1]. A language is considered object-oriented if it is object-based and also has the capability of polymorphism and inheritance. Metrics are standards that define measurable attributes of entities, their units and their scopes. Metrics are the essential building blocks of any evaluation process, since they allow the establishment of specific goals for improvement [3].

The software agent methodologies are PROMETHEUS, GAIA, OperA, O-MaSE, Tropos and GORMAS[11][7]. GAIA is a methodology for agent-oriented analysis and design. Prometheus supports the design of agents that are based on goals and plans. OperA model describes a MAS as an organizational structure. To examine the quality of the software agent, the properties measured are Social ability, Autonomy, Pro-activity, Reactivity, Adaptability and Mobility.

2. BACKGROUND WORK

To measure the metrics and evaluate the characteristic quality of an agent in a MAS and the development methodologies are discussed below

2.1 Specification of Normative Environment with the Support of Agent Methodology [7]

There is an increased collaborative work with decentralization process in various processes which leads to a demand for flexible, adaptive system in large-scale to support interactions between institutions and people in heterogeneous environment. In a normative environment the actors contains certain duties, rights and restrictions to be followed and fulfilled. A software system's final design is determined by a normative environment so it must be considered an important issue at the system design.

In many domains like health and commerce there is a demand for dynamic, complex and decentralized system in which interactions between various entities and institution takes place to exchange services to achieve the objective. These entities are mostly heterogeneous and autonomous also they are not bound to any central authority that govern them all together also neither of them have authority on each other behaviour. To achieve interaction and stability in system the behaviour of the entity must be known. Usually these domains are regulated under government legislation and internal legislation of the institutions in the system. Regulated system with dynamic regulation in social environment are known as normative environment. Normative context of any organization defines the set of norms that affect only those entities that are part of the organization.

Example : mWater case study
mWater is a decentralized framework where the users with the water rights are allowed to trade voluntarily their rights with other users by fulfilling certain pre-established rules.

2.1.1 Requirements for Measuring the Normative Environment

In a normative environment the entities behaviour and relationships are restricted to a set of norms. The needs for designing normative system is analysed from a semantic perspective. The rules that regulate the behaviour of a specific institution and/or group of entities are known as institutional norms. The different entities are bound to contracts that are flexible and expressive these entities that play a role in such contracts are known ah play role contracts. Structural norms defines the structure of the overall system that is the relationship between the organization within it and the system roles. The various methodologies are analysed by the criteria defined in this paper.

The analysis of the state shows there are few mature agent methodologies that are used in developing normative systems. The combination of the partial solutions to obtain complete development methodology that a developer can use is not an easy work since each approach uses different semantic, terminology and metamodel constructions. If methodology don’t include methods to verify coherence of different norms of system then the developer must verify it manually hence it is a complex task due to the huge number of norms that should be taken into account. The behaviour of the entity in a system must be internally specified in implementation of entity. There is lack of quantitative metrics to compare with various methodologies since metrics are the essential building block of any agent system.

2.2 Suitability Assessment Framework [11]

The intelligent architecture is Multi Agent System. For creating Multi Agent System (MAS) Agent Oriented Software Engineering (AOSE) is supported [11]. But there arises two questions that is the first is how do we determine the suitability of MAS implementation of a particular problem and the next is how this can be done without an AOSE expert? The objective of this paper is that the engineers can evaluate the MAS technology without an AOSE expert. The key criteria for identifying the MAS solution can be based on the iterative process. The paper mainly focuses on the MAS Technology. They mainly concentrate mainly on engineers rather than the AOSE expert. The paper was mainly illustrated on two real world problems for accessing the suitability of Multi Agent System.

This paper mainly concentrates the MAS features for determining the suitability. In this paper they have implemented on different environments such as uncertain environment, dynamic environment, dependable environment, open environment, distributed environment. The properties include communication quality, efficiency, robustness, reliability, flexibility, responsiveness, indeterminate, concurrency, scalability, legacy and other interactive types. They are validated against the existing framework. Based on this the architecture for MAS have been developed. The interaction types are based on simple enquiries, negotiation, cooperation, argumentation, complex interaction. Each author has validated based on these properties and the values are represented based on each properties and the environment. The work is mainly based on the assessment of the framework for the Multi Agent System (MAS) [5].

The paper has the key features of the different problem domains. They are iteratively featured based on the significant domains of whether MAS is suitable or not. It access the suitability framework of the MAS solution for the set of problem related issues. The conditions have proved that the software engineers can evaluate an MAS application without an AOSE expert. The framework is always independent and they can be applied to any scenario for MAS to determine the suitability. In this system they allow the designer to consider the appropriate of MAS solution for satisfying certain conditions. They framework is for on-going validation. We apply our proposed framework for refining the framework as the description and the associated ratings and this ensures that they didn’t miss any of the key problems. This is implemented on other systems and MAS validation is measured. This can be further provided by examining the consistency of the rating between the different raters and the software engineers to perform the ratings.

2.3 Agent Oriented Program towards Multi-Paradigm Metrics[4]

The multi-paradigm metrics is applicable to both object oriented and agent environment in which they are suitable. The multi agent system is evaluated quantitatively. The concrete paradigm is used for coupling and cohesion.

The metrics are available for the object and it is realized using object oriented programming language. The metrics for agent is done using agent oriented programming language. The performance cost for object is lesser than agent. The combination of factors, characteristic and attribute is known as quality. The factors are functionality, reliability, usability, efficiency, maintainability and portability. The combination of multiple metrics is not possible. The metrics measure a single quality attribute.

The focus is on maintainability and it is an interesting potential benefit of agent based software. The maintainability is based on algorithmic complexity, structural or design complexity and size. There is lot of measures for size and algorithmic complexity. The measures for structural complexity are not sufficient. The sub factors of structural complexity such as coupling and cohesion are dominant. The structural metrics is to predict the bugs in present in future. The structural complexity used to develop higher quality software.

The goal question metric is employed. Identical solution is analysed for the problem using different technology. The solution is compared using suitable metric and these methods are used widely. For the rule based program, knowledge based- system and concurrent logic the structural metric is proposed. The evaluation metric for the aspect oriented programming is developed. Automated method is used for measuring the information for two digital artifacts. The metric is related directly to the independent path for a given problem.

The architecture used has two types of agent vacagent and boss agent. Each vacagent directly control the each vacuum robot. The boss agent is not situated in the environment. The boss agent receives the message from other agent.

The structural metric must to gather manually. There is lot of programming language for high level design for understanding and reuse of abstraction and it supported by language construction. The abstraction is realized as program source code and it is a executable one. The element is composed of many elements and it is used to form a aggregation hierarchy. There will be many changes in the software during the lifetime of the software. The existing software design is predicted using the structural maintainability. There will be risk and lot of effort for the changes. The focus is mainly on modification of the existing program features. The possible refactoring available are infinite. The useful insight is proposed in the scope and changes impact and the availability of code segment. The object is coupled to another object and each of the object is similar. The action is defined as the method of one class modifying the other and the access of other is also possible. The rename operation is on the left right class. The internal action is done by the vacant. The plans must to be named if they did not then they did not meet out definition. Coupling between the elements is count the number of class to be coupled. The class cannot be coupled to itself. Lack of cohesion element measures the similarity of the class by comparing the instance variable of the method. The metrics is a tool for the evaluation of the future problems which are going to be arises and it is a increasing technology. There are many unappealing property and they are pair wise combination in class. The version is not sensitive to dependencies between the methods.

In multi-paradigm is a substantial functionality postponed to object oriented element. The paradigm independent coupling and cohesion metrics are proposed. These metric is evaluated theoretically and the validity predict the maintainability. The rename factor is not proved in the proposal. The CBO and LCOM are well validated and they are theoretically proved. The dataset required for validation of CBO and LCOM are greater. Independent experimental validation is not done and it left for the future work. The agent peak programming is uncovered in the proposed metric.

The plan must to be named and naming the plan is optional in agent speak and it must to be named where it is possible and it used to locate the code issues. The agent possible mental states is determined easily using belief and goal. The maintainability is measured effectively rather than decomposition and reuse. The experimental validation of the CBE and LCE metrics are available. Model which is under lying to the programming language should also be applicable for the evaluation of the multi paradigm. Examination of the syntax and structure of the goal and the program must suggest the model is comfortable and some interesting must to be included. The goals include the terms, macros, prediction and action ,the realization of these factors are short and many element must to be completely defined. The program in goal has many element and it could lead to changes in CBE and LCE .

Dependency information gathered contain only a small proportion if reflection from the CBE and LCE. There are different contribution to the coupling and cohesion. The software maintainability is applied are not is debated .The relative dependency is important and intuitive. Multiply interdependent will require less future maintenance only when the dependency are collocated. If the new metric has maintainability then it leads to recommendation of multi-paradigm program and language. Visualizing and summarizing the raw dependency data in flexible means will generate the new metrics. These type of tools will allow the researchers to concentrate on dependency and types to current context.

2.4 Social Ability of Software Agents [9]

A software agent is evaluated by measuring its characteristic quality by using a set of measures for those features. Since these set of measures are not yet being developed for testing agent-oriented software. Few software measures are adopted from object –oriented paradigm due to the common features between them. The ability of a software agent to interact with other agents to achieve its goal is characterized as social ability property. This paper gives a set of measures for evaluating the software agent’s social ability.

The measures are divided into three types

- Syntax-based
- Execution based
- Objective based

The measures derived in this paper belong to Execution based or dynamic based. To validate the results in the quality of a software agent the evaluation process must be conducted in a controlled environment called as benchmark. The measures are stated in means of formula which expresses the measure as a function of one or more parameters. The result of each measure is normalized in the interval [0,1].

2.4.1 Measures of the Attributes of Social Ability

The social ability property has three main attributes

- Communication
- Cooperation
- Negotiation

Communication

This attribute is measured using the following metrics

a *Response for message(RFM):*it measures the number of messages sent in response to a received message from an agent its value increases as the agent becomes more communicative.

b *Average message size(AMS):*It measures the data size of the message sent or received by the agent during its communication if message size is too large it leads to bad communication

c *Incoming messages(FIM):* It measures the number of incoming messages to an agent during its lifetime, higher values shows that the agent has depended agents that requires its services.

d *Outgoing messages(FOM):*It measures the number of outgoing messages from an agent during its lifetime, if its value id high it shows that the agent is a dependent agent.

Cooperation

The attribute is measured using the following metrics

a *Service request rejected by the agent(SRRA*):It measures the percentage of the rejected agent services under cooperation. The measure is considered optimum when the percentage of rejected services are low.

b *Agent services Advertised(ASA*):It measures the total number of services that an agent advertise on its yellow page directory. A low value indicates that the agent is less cooperative.

Negotiation

The attribute is measured using the following metrics

a *Agent goals achievement(AGA):*It measures the agent's negotiation to accomplish its goals

b *Messages by a requested service(MRS):*It measures the number of messages sent and received with the other agent during its negotiation process with the agent requesting service form it.

c *Messages sent to request a service(MSS):*It measures the number of messages sent or received by the agent during the negotiation process with another agent when the agent is requesting service from the other agent.

Various metrics for the attributes of an agent-oriented system is defined that measures the quality of the characteristic of the software agent. Still more accurate metrics must be derived to measure the quality of the software agents characteristics. A quality evaluation model that evaluate the models across various software agent applications considering the different properties and attributes the agent posses must be designed and developed.

2.5 Reactivity Property of a Software Agent [10]

The agent technology uses the complex developing the distributed applications. The software agent are the building blocks of Multi Agent System (MAS).Software agent are different so that they possesses different properties such as pro- activity, reactivity[10], social ability, autonomous, and efficiency. Agent behaviour might differ from different inputs and thus for evaluating the software agent quality is difficult process. The quality of characters and the agent behaviour are not full provided. This paper mainly deals with the reactivity paper where quality of the software agent is measured. The software agents are defined as the abstraction of computer program that acts on behalf of the another. There is lacking in the testing mechanism for the software agent based system. The reactive property is defined as the how it responses to the changes that occur in the environment. Then the other properties also include autonomous which means the some of the self control are performed over the states and the actions. In this paper the quality of the software agent is tested. The reactive property also describes how the agent perceives through the environment and in timely manner the changes to occur.

The measures are not proper for the reactive properties. The attributes include

- Interaction level
- Communication level
- Perception level

There are different levels in each metrics and the Interaction level includes

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

The metrics in the communication level are

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

The metrics in the perception level are

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

They are implemented in JADE. The case study is performed for the online shopping system.

The software quality is measured for the reactivity property. They are simple that they can be based on metrics. The attributes are measured and some more metrics can be added for each attribute so that the attributes and the property more accurately measured.

2.6 Software Agent Pro-Activity [8]

There is different property for evaluation the software agent. The properties used to evaluate the agent are social ability, autonomy, pro-activity, reactivity, adaptability, mobility. The pro-activity which is related to agent goal directed behaviour is evaluated using the pro-activity. The agent ability to interact with the other agent is known as the social-ability. The agent is able to perform any operation without the need of any human help is known as autonomy. The must to be able to adapt to the environment and the agent must to be flexible is known as ad-aptability. The agent is able to move in any environment by itself and it must to be interact with the new environment to gather the information. The agent is able to react to the perceived action is known as reaction. The agent must take the initiative to satisfy its goal. There are different attributes used to evaluate the agent goal directed characteristics. The pro-activity is the most important one in the software agent because for each every work there must to goal.

The measures which are available are taken from the procedural and object oriented paradigm. There is only a few evaluation processes for the software agent. Software agent pro-activity is one of the most important one in the goal directed and initiative. The agent must take the initiative rather than performing simple work in unpredictable environment. There are several studies related to pro-activity but that study is not related to the evaluation process. The measures which are all used are to reach the result. Several studies are there for the software agent and these studies are based on the quantitative methods, procedural and object oriented paradigm.

There are different attribute for pro-activity property they are initiative, interactive and reaction. Each and every attribute has different metrics. The -initiative is the software agent ability to attain the goal and the agent must take the initiative or action to attain the goal in any environment. The initiative attribute consists of three metric they are number of roles, number of goals, messages to achieve goals. The number of roles is to measure the potential roles the software agent performs. The number of goals is to measure the total number of goals which are all achieved by the software agent. Messages to achieve goals are used to measure the agent initiative to attain its destiny.

The interaction attribute is that the software agent ability to interact with the other agent in any environment. The interactive attribute can be measured using these metric they are services per agent, number of messages types. The service per agent measures the interaction for the number of service request. The number of messages type measures the agent interaction with the different type of messages received and the agent can respond.

The reaction attribute is the software agent capability to react to the other agent in any environment according to the response behaviour. Each and every measure is stated using the formula. The reaction attribute consists of two metric for the evaluation of the agent and they are number of processed request, agent operation complexity. The number of processed request measures the agent ability to respond for the number of received and resolved request and this is done during the execution. The agent operation complexity is used to measure the complexity that the software agent faces before attaining the goal. The software developers can use any complexity measures.

The different attribute for the software agent has been evaluated. These attribute which are all used are not sufficient for the evaluation of the software agent goal directed behaviour known as pro-activity. The quality evaluation model is not done for the evaluation of software agent. These metrics are not sufficient for the evaluation of the software agent. The software agent must to be evaluated in deep so then only agent characteristic can be in proper manner. There are several characteristic are must to be added for agent goal directed behaviour. The existing metrics for the properties are less sufficient for evaluating the software agent. The attribute which are used to measure the software is less sufficient. The agent must to be evaluated to the leaf for the proper evaluation. The more number of evaluation metric must to be included for proper evaluation of the software agent.

3. CONCLUSION

In a normative environment there is high demand for the flexibility and adaptability for the interaction purpose. The entity behaviour must be known for the proper interaction but there is lack of quantitative metrics to compare the various methodologies. The MAS features are determined for the suitability of the environment in which the agent is going to interact. Since the system is implemented on different environment the consistency of rating must be examined. The CBE and LCE metric reflect only small maintenance and there is less future maintenance.

Reviewing the literature the existing measures for evaluating the software agent is insufficient. The communication level and the goal directed behaviour are not evaluated to the fullest. The new metrics are to be included to evaluate the agent system quality .Since the existing metrics for the properties of an agent system are less sufficient to prove that characteristic of the agent in a MAS. The more accurate measures are needed to evaluate the reactivity property. we have suggested certain new metrics so that along with the existing metrics the new metrics must to be added to measures the agent’s property more accurately.

REFERENCES

[1]. Bimlesh Wadhwa, Koh Shin Jang and Teo Ee Nam “Object and Agent Metrics Approach” International Journal of Engineering Science Invention Research & Development; Vol. I Issue II August 2014.

[2]. Dino Franklin and Ana Abrao “Measuring Software Agent’s Intelligence”

[3]. Prof Dr Erika Horn, Mario Kupries, Dirk Glode “Properties and Models of Software Agents and Agent Systems”1998

[4]. Howell R. Jordan and Rem Collier “Evaluating Agent-Oriented Programs: Towards Multi-paradigm Metrics”

[5]. Christos Dimou, Andreas L. Symeonidis, and Pericles A. Mitkas “Evaluating Knowledge Intensive Multi-agent Systems”

[6]. Chrysanthi E. Georgakarakou & Anastasios A. Economides “Software Agent Technology: an Overview Application to Virtual Enterprises”

[7]. Emilia Garcia, Simon Miles, Michael Luck, Adriana Giret “Evaluating how agent methodologies support the

specification of the normative environment through the development process" Springer US 2014

[8]. Fernando Alonso, José L. Fuertes, Loïc Martínez, and Héctor Soza "Measures for Evaluating the Software Agent ProActivity " Software Engineering Advances (ICSEA), 2010 Fifth International Conference on 22-27 Aug 2010 Page(s): 319 - 324

[9]. Fernando Alonso, José L. Fuertes, Loïc Martínez, and Héctor Soza "Measuring the Social Ability of Software Agents" Software Engineering Research, Management and Applications, 2008. SERA '08. Sixth International Conference -on 20-22Aug 2008 Page(s) 3-10

[10]. N.Sivakumar and K.Vivekanandan "Measures for Testing the Reactivity Property of a Software Agent" (IJARAI) International Journal of Advanced Research in Artificial Intelligence,Vol. 1, No. 9, 2012.

[11]. Ghassan Beydoun, Graham Low, Paul Bogg "Suitability Assessment framework for agent-based software architecture"