ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2013.0085

A QUANTITATIVE FRAMEWORK FOR EARLY PREDICTION OF COOPERATION IN MULTI-AGENT SYSTEMS

S. Ajitha[1], T. V. Suresh Kumar[2] and K. Rajanikanth[3]

Department of Master of Computer Applications, M. S. Ramaiah Institute of Technology, India

E-mail: [1]ajithasankar@gmail.com, [2]tvsureshkumar@msrit.edu and [3]rajani341949@yahoo.com

Abstract

A Multi-Agent System (MAS) is a system composed of multiple interacting intelligent agents. MAS can be used to solve problems that are difficult or impossible for an individual agent to solve. The different characteristics of MAS help in solving highly complex distributed problems. One of the important characteristics of MAS is its cooperative nature. This character helps different agents to interact with each other by exchanging messages. One of the major challenges in MAS is quantifying the cooperation between agents. This paper presents a framework for the quantification of cooperation between agents in MAS. We propose a methodology which helps to quantify the cooperation in the early stages of software development using a UML sequence diagram and a mathematical model. The proposed techniques are illustrated with the help of a case study. The numerical results we got were quite satisfactory.

Keywords:

Multi-Agent System, Performance Prediction Process Model, UML, Agent Characteristics, Software Performance Engineering

1. INTRODUCTION

Over the last decade the popularity of agent-based systems has increased rapidly because agents bring intelligence, reasoning and autonomy to software systems [1]. Agents are used in an increasingly wide variety of applications from simple e-mail filter programs [2], to complex mission control and safety critical systems including air traffic control, such as OASIS [3]. Cooperation is often presented as one of the key concepts which differentiate multi-agent systems from other related systems such as distributed computing, object-oriented systems, and expert systems [4]. The agents in MAS cooperate with each other in the system in some way to accomplish some goal. Such cooperation can be communicative in that the agents communicate with each other by sending and receiving of signals/messages. Another important characteristic of MAS is that the environment is no longer static. As opposed to single agent systems, the dynamics of the environment are affected by the actions of other agents in that environment.

Cooperation is the fundamental characteristic of MAS where the overall MAS exhibit significantly greater functionality than the individual agents. It has received a considerable amount of attention in the MAS literature. Many classical theories are applied in the research of cooperation via logic theory, game and economic theory and Petri-net etc. A series of cooperation models were put forward based on these theories [5, 6, 7, and 8]. As cooperation has attracted so many researchers, many applications have emerged in different areas [9]. There are several agent oriented software engineering methodologies, architectures available and few are MESSAGE, TROPOS, RETSINA [10, 11, 12]. In our studies we have considered the RETSINA architecture [13]. This architecture consists of three different types of agents namely Interface Agent, Task Agent and Information agent. Interface agents interact with the user for their specifications and deliver results to the users. Task agents formulate the plan and carries out the specified tasks with the help of other task agents and information agents. The Information agents collect the information needed for the task agents from different information sources.

In this paper we are proposing an approach to find the cooperative index (CI) of agents in MAS at the early stages of the SDLC. We designed a mathematical model to predict the CI value at the early stages of software development. Our methodology has two options to calculate the CI value, UML sequence diagram; mathematical approach. The remaining parts are; Section 2 presents related works in the area of cooperation in MAS. The proposed model for quantifying cooperation between agents is addressed in section 3. This section consists of representation of static, dynamic models for CI and a methodology for predicting the CI of static and dynamic models. Section 4 is presents the UML models of MAS. Section 5 presents the output of the numerical analysis. Section 6 presents the conclusion and future work.

2. RELATED WORK

Nicholas R Jennings, Katia Sycara and Michael Wooldridge in [14] provided an overview of research and development activities in the field of autonomous agents and multi-agent systems. D. I. Mark in [15] has proposed a mathematical and computational aspect of the social reasoning process of agents in MAS. The author defined an abstract representation of cooperation structures, and investigated the question of whether or not cooperation is feasible with respect to an agent's goal, and reported that the answer is an NP-complete problem. However the problem of computational complexity and tractability has been overlooked in the design of MAS. In [16] Ping Wang has made a survey in cooperation in MAS. He observed that cooperation is related to the interactions among agents, and it is very complex to understand, describe and realize. Different theories are applied to the research and various models are proposed. The typical theories applied to the research of cooperation include game and economic theory, logic theory, Petri-net theory and software engineering. Models based on these theories help us to understand, describe and realize cooperation. In [17] R. G. Smith and R. Davis have suggested two complementary forms of cooperation in distributed problem solving: task-sharing and result-sharing. These forms are useful for different types of problem and for different phases of distributed problem solving. RoieZivan et al. in [18] proposed a paradigm for multiple agents to solve a distributed problem, acting partly cooperatively and keeping a limited form of self-interest. The design goal of the paper 'Concenus and cooperation in Networked Multi-Agent Systems' [19] is to execute tasks cooperatively exercising both the decision making and control capabilities of the vehicles. In real life networked multi vehicle systems there are a number of limitations including limited sensing capabilities of the vehicles, network bandwidth limitations, as well as interruptions in communications due to packet loss and physical disruptions to the communication devices of the vehicles. In this paper the author discussed a theoretical framework for analysis of consensus algorithms for networked multi-agent systems with fixed or dynamic topology and directed information flow.

Agent cooperation is one of the well studied areas of MAS. Agent cooperation is beneficial whether the problem is difficult or easy. G. Adel et al. in [20] proposed a cooperative game theory (CGT) for coalition formation in multi-agent systems, where a novel model for the cooperative game has been used. The implementation of cooperation is enforced in terms of coalition formation and algorithms for their formations are discussed. However a quantified approach for cooperation may be useful in experimenting for coalition formation. R. L. Victor in [21] explains the cooperative MAS. In this paper the author discussed the application of MAS, the nature of MAS interaction and major challenges and research direction. C. Gutierrez and I. Garcia-Magarino suggested a metric suite for the communication of MAS in [22].

There are works related to the use of MAS to achieve high levels of QoS where the system was designed to support resource allocation in cellular data services in such a way that it meets both customer satisfaction and cost effectiveness. The solution by the authors was to design agents within three modules built into the scheme: the knowledge source, the blackboard system and the control engine. The response time is used as an indicator of the QoS. And the approach suggested improved the QoS by means of measuring and improving the communications policy. However the early prediction of response time is a major research challenge. G. Wojeiech, K. Halina in [23] introduced a solution for coalition formation problem in MAS based on an evolutionary algorithm for solving assumed tasks.

K. Kavi et al. in [24] used Unified Modeling Languages (UML) for modeling and design of Multi-Agent Systems. They presented a framework for modeling, analysis and construction of agent-based systems. The framework is rooted in the Belief Desire Intention (BDI) formalism and extends the UML to model MAS. Use Case Goal Diagram to model the relationships between use cases and goals; Agent Domain Model to facilitate understanding of domain knowledge of an agent; Agent Sequence Diagram to model interactions within an agent. Similarly, Agent Activity Diagram and AgentState chart Diagram are introduced. This [25] paper lays out many problems associated with the design of an agent architecture which has to operate in an open and large scale multi-agent environment. A method for addressing these problems and a generic architecture based on this approach was discussed. The presented architecture has five components: local agent scheduling, multi-agent coordination, organizational design, detection and diagnosis and on-line learning that is designed to interact so that a range of different situation specific coordination strategies can be implemented and adopted as the situation evolves. The proposed work cannot be treated as a finished work but rather as an instance of how a wide range of ideas developed by the Multi agent/DAI community can be integrated into a viable computational framework. Before deciding on the level of a particular feature for example, multi-Agent cooperation an indicator is required for incorporating non-functional requirements.

Many authors defined cooperation and cooperation structures, with the help of game theory models. Also from literature we identified that most of the work on cooperation has been done based on formal languages which give a theoretical approach for cooperation. Quantifying the cooperation between agents is not addressed in the literature. Cooperation between agents takes place with the help of message communication. Cooperation between agents with the help of messages communicated is not addressed in the literature. The models derived from mathematical models are quantitative rather than symbolic. So we are proposing a mathematical model which quantifies the cooperation between agents in MAS. At early stages of SDLC we have used use case diagram to identify the number of agents in the system and overall scenario of the system. The interactions between agents are addressed with the help of UML sequence diagram.

3. PROPOSED MODEL FOR QUANTIFYING COOPERATION BETWEEN AGENTS

Cooperation in MAS enables groups of agents to solve problems effectively through the neighboring agents which can help them to perform specific tasks. The complexity involved in making these decisions can be seen in a situation, where one agent needs the result of a sub problem from another agent. This result has to be communicated to the main agent for executing the responsibility of that agent. Also if one agent is not able to perform the task it forwards the task to its neighbor who can accomplish this task.

Cooperation between agents is defined as how effectively agents respond to the request of its neighbor agent. We have modeled the scenario of cooperation of agents with its neighbor by message passing. We have defined a term called cooperative index (CI) which quantifies the cooperation between the agents by considering the number of messages an agent received from the neighbor, the number of messages forwarded by the agent to the neighbor, the total number of messages generated by the agent for accomplishing its own task and the number of messages sent out of the total number of messages generated to accomplish the agent's own task. We have formulated this model by considering models developed in [26, 27].

The cooperation in the context of MAS is difficult to predict in the early stages of development of the system. Hence for predicting the number of messages communicated between the agents, we have considered probability distributions. The scenario we considered for cooperation includes the number of messages sent, forwarded, generated and received. We have used the exponential distribution for forwarding and receiving the messages because, in the literature exponential distribution are widely applied in generating arrival patterns. When the probability of occurrence is small then the distribution of events can be approximated by the Poisson distribution. When we consider the cooperative nature of agents self messages generated by agents will be less hence we applied Poisson distribution for total messages generated and number of messages sent by an agent for its self processing tasks.

A software application which works based on MAS theory, the cooperation may be in terms of sharing the messages across the network, resource utilization, sharing of knowledge etc. The resource utilization for the cooperative nature of the agent is calculated with the help of the variable R_u. We have used the value of R_u in the range of 0 to 1. To quantify this aspect we have defined a new formula for accessing the cooperative index between the agents. Associated to every resource there exists a constant $0 < R_u < 1$ defining the importance given to a particular resource. If $R_u = 0$, the agent can provide the resource which are accessible to it for other agents. This means the agent is using specific resources for generating/self processing of the messages so it can share its resources to other agents. When $R_u = 1$ the cooperative index value is high which claims the usage of resource is very high.

Let us assume MAS is formed by the coalition of n agents. The agents are numbered as a_1, a_2, …, a_n. The Fig.1 gives the view of the MAS as a block diagram. Let us consider an agent a_i and its neighbor a_j. Their communication can be diagrammatically represented as in the Fig.2.

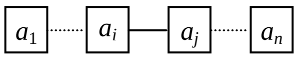

Fig.1. Block Diagram of MAS

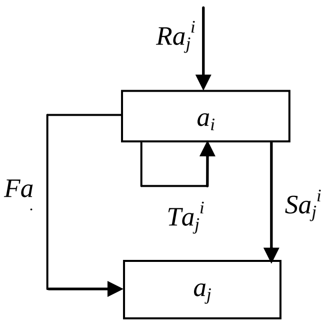

Fig.2. Block Diagram of Communication of Agents in MAS

3.1 STATIC SCENARIO FOR COOPERATIVE INDEX

We define W_{ai} as the measure of messages forwarded by 'a_i' in terms of the ratio between messages that neighbor 'a_j' forwarded after a request by 'a_i' or received by 'a_i' as the final destination and self interested sent messages by 'a_i' for processing.

$$W_{ai} = \begin{cases} W(i) & \text{if } Sa_i + Fa_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where,

$$W(i) = \frac{\sum_{j\in N_i} Fa_j^i + Ra_j^i}{Sa_i + Fa_i} \tag{2}$$

Sa_i = The number of messages 'a_i' sent to its neighbor for accomplishing its own task.

Fa_i = The number of messages 'a_i' forwarded to its neighbor agents (an agent forwards to other agents if it cannot accomplish the task).

N_i = Neighbor of 'a_i'

Fa_j^i = The number of messages 'a_j' forwarded to the 'a_i'

Fa_j^i = The number of messages 'a_j' received from 'a_j'

We define G_{ai} as the ratio between the sent messages for processing by the agent itself to the total messages agent wanted to send for self processing.

$$G_{ai} = \begin{cases} g(i) & \text{if } \sum_{j\in N_i} Ta_i^j > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where,
$$g(i) = \frac{\sum_{j\in N_i} Sa_i^j}{\sum_{j\in N_i} Ta_i^j} \tag{4}$$

Fa_i^j = The total number of messages 'a_j' generated for sending to its Neighbor 'a_j' for self processing

Sa_i^j = The total number of messages 'a_j' sent to its neighbor 'a_j' for self processing

3.2 DYNAMIC SCENARIO FOR COOPERATIVE INDEX

In dynamic model we have considered the behavior of the system by considering different time intervals. These intervals are taken into consideration while applying the above formula.

We defined $W_{aj}(t_k)$ as the measure of messages forwarded by 'a_j' at time t_k expressed in terms of the ratio between messages that neighbor 'a_j' forward after a request by 'a_j' or received by 'a_j' as final destination and sent messages by 'a_j' for self processing.

$$W_{ai}(t_k) = \begin{cases} W(k) & \text{if } sa_i(t_{k-1}) + Fa_i(t_{k-1}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where,

$$W(k) = \frac{\sum_{j\in N_i(t_k)} Fa_j^i(t_k) + Ra_j^i(t_k)}{Sa_i(t_{k-1}) + Fa_i(t_{k-1})} \tag{6}$$

$Sa_i(t_{k-1})$ = The number of messages 'a_i' sent to its neighbor for its self process during the time interval (t_{k-1})

$Fa_i(t_{k-1})$ = The number of messages 'a_i' forwarded to its neighbor during the time interval (t_{k-1})

$N_i(t_k)$ = Neighbor of 'a_i' during time interval (t_{k-1})

$Fa_j^i(t_k)$ = The number of messages 'a_j' forwarded to the 'a_i' during the time interval (t_{k-1})

$Ra_j^i(t_k)$ = The number of messages 'a_j' received from the 'a_i' during the time interval (t_{k-1})

$Wa_i(t_k)$ is defined as the ratio between the sent messages for processing by the agent itself to the total messages agent wanted to send for processing at time t_k.

$$G_{ai}(t_k) = \begin{cases} g(t_k) & if \sum_{i\in N_i t_k} Ta_i^j t_k > 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where, $$g(t_k) = \frac{\sum_{j\in N_i(t_k)} Sa_i^j(t_k)}{\sum_{j\in N_i} Ta_i^j(t_k)} \quad (8)$$

$Ta_i^j(t_k)$ = The total number of messages 'a_j' generated for sending to its neighbor 'a_j' for self processing during the time interval t_k

$Sa_i^j(t_k)$ = The total number of messages 'a_j' sent to its neighbor 'a_j' for self processing during the time interval t_k

3.3 AN ALGORITHM FOR COOPERATIVE INDEX

The following steps provide an insight to the algorithm for quantifying the cooperative index (CI). The algorithm proposes two different approaches to find the CI of agents in MAS. One approach is to calculate the CI value of an agent from the UML diagrams and the second approach is to calculate the CI value without using design diagrams. In the case of calculating CI from UML diagrams we used two diagrams namely use case diagrams and sequence diagrams. From use case diagram the overall scenario of the system can be identified. It shows the number of agents in the system the different use cases and the associations between them. The sequence diagram is used to calculate the CI of each agent. In the second approach we used probability distribution to get quantified results to our approach. Here we assumed that MAS is already constituted with the coalition of n different agents. When an agent needs cooperation it identifies its neighbors for communication. Once the neighbors are identified some agents receive messages and some agent forward messages to other agents. The exponential distribution is used for the variable Ta_i^j and Sa_i^j . The Poisson distribution is used for the variable Ra_i^j and Fa_j^i. W_{ai} gives the total number of messages forwarded by an agent on behalf of the requests it received from another agent to its neighbors and G_{ai} is the total number of messages an agent sends to its neighbors for self processing. Here R_u represents the percentage of resource used by the agent while processing the messages.

Begin
{
Consider the architecture of MAS as RETSINA
While (*calculate CI using Design Diagrams*)
{
Identify the Number of agents and use cases.
Develop a Use Case Model for the given MAS application.
Analyze the interaction between agents.
Draw the Sequence Diagram.
Calculate the number of forwarded messages and received messages by an agent.
Calculate the total number of messages generated for self process and total number of self messages send by an agent.
Let R_u*= the usage of the resource for the message transaction.*
If (*Static behavior*)
{
Calculate W_{ai} *and* G_{ai} *using Eq.*(1) *and Eq.*(3).
Calculate CI of Agent $a_i = R_u * W_{ai} + (1-R_u) * G_{ai}$
Else
{
Calculate $W_{ai}(t_k)$ *and* $G_{ai}(t_k)$ *using Eq.*(5) *and Eq.*(7)
Calculate the CI of agent $i = R_u*W_{ai}(t_k)+(1-R_u)*G_{ai}(t_k)$
}
}
Consider a MAS formed by the coalition of n Agents $a_1, a_2, \ldots a_n$.
Identify the neighbour's for an agent a_i
If (*the scenario is static*) *then*
{
Identify the number of agents in the system
Identify the neighbour's for an agent a_i
Ta_i^j = *a random number using suitable probability distribution.*
Ra_i^j = *a random number using suitable probability distribution.*
Fa_i^j = *a random number using suitable probability distribution.*
Sa_i^j = *a random number using suitable probability distribution.*
Calculate W_{ai} *and* G_{ai} *using Eq.*(1) *and Eq.*(3)
Let R_u = *the usage of the resource for the message transaction.*
Calculate CI of Agent $a_i = R_u * W_{ai} + (1-R_u) * G_{ai}$
}
Else
{
Let the time interval be t1, t_2, ----, t_k, -------, *tn*.
Identify the neighbour's for an agent a_i *at time* t_k *say* $a_i(t_k)$.
$Ta_i^j(t_k)$ = *a random numberusing suitable probability distribution.*
$Ra_i^j(t_{k-1})$ = *a random numberusing suitable probability distribution.*
$Fa_j^i(t_{k-1})$ = *a random numberusing suitable probability distribution.*
$Sa_i^j(t_k)$ = *a random numberusing suitable probability distribution.*
Calculate $W_{ai}(t_k)$ *and* $G_{ai}(t_k)$ *using Eq.*(5) *and Eq.*(7).
Let R_u*= usage of the resource for the message transaction.*
Calculate the CI of agent $i=R_u*W_{ai}(t_k)+(1-R_u)*G_{ai}(t_k)$
}
}
End

4. UML MODELS OF MAS

UML can be used to design a Multi-Agent system (MAS) at the agent level. The MAS we have considered is based on supply chain management (SCM) systems. Agents are autonomous and can operate in open electronic environments that are now becoming very popular. Agent technology allows software engineers to develop solutions which can co-exist and operate along with external and legacy systems. This is important in the context of SCM that involves many parties which might use different technologies. The multi-agent approach is “a natural way to modularize complex systems” [28] which is the case with SCM. This approach allows separating different tasks within the SCM and exploring them both independently and in relation to each other. The system can be broken down into separate building blocks, each concentrating on a particular part of the supply chain. By replacing one building block with another and by combining them in different ways, various versions of the system can be created. In this way, the influence of changes in behavior in each link of the supply chain can be thoroughly studied.

The proposed MAS include five agents corresponding to each entity in the supply chain (Supply Agent, Inventory Agent, Production Agent, Delivery Agent, Manager agent). While following their own goals, the agents work in cooperation in order to achieve the common ultimate goal to maximize the overall profit. This goal can be split into the following sub-goals, maximize sales revenue, minimize component purchase prices, minimize component and product holding costs and minimize penalties for late delivery. Each agent in the system is responsible for one or more of these goals. The use case diagram for the system we have considered is represented in the Fig.3. This diagram helps to identify the number of agents in the system and its association with other agents. Thus this diagram represents an overall scenario of the system.

The sequence diagram is represented in Fig.4. Sequence diagrams are used to represent the interaction between agents. Here to represent the dynamic behavior we have divided the time slots into three divisions as T1, T2, and T3 and represented in the Fig.4.

This helps us to find the interaction between agents at each time interval. From this diagram how we identified the types of messages exchanged between agents in each time interval and the calculation of the Cooperative Index at the early stages of software development is discussed. Here we have considered five agents namely Manager Agent (MA), Production Agent (PA), Interface Agent (IA), Supply Agent (SA) and Delivery Agent (DA). We have used UML stereotypes to represent the agents and the message transfer between agents.

From the figure at time T1: <<MA>> received one message, forwarded one message, generated two self processing messages and send one message for self processing. The agent <<PA>> received one message and forwarded one message, no self processing messages for this agent at time T1. The <<IA>> agent received two messages and forwarded two messages. The agent <<SA>> received one message, forwarded one message; self processed one message and sent one message for self processing. Similarly we can identify the messages for each agent at time intervals T2, T3. By using the Eq.(5) and Eq.(7) we can calculate the no of messages an agent forwarded for other agents ($W_{ai}(t_k)$) and messages an agent sent for self process ($G_{ai}(t_k)$). From this diagram for agent <<MA>> we calculated (W_{aMA}(T1) as ‘2’ and (G_{aMA}(T1)) as ‘5’. By using the formula for calculating CI of agent “agent $i = R_u * W_{ai}(t_k) + (1-R_u)*G_{ai}(t_k)$”. We calculate the cooperative index of agent <<MA>> as 2 by considering the resource utilization as maximum. We can substitute different values for resource utilization and a range of CI values can be calculated. Similarly for other agents at different time intervals we can calculate CI values.

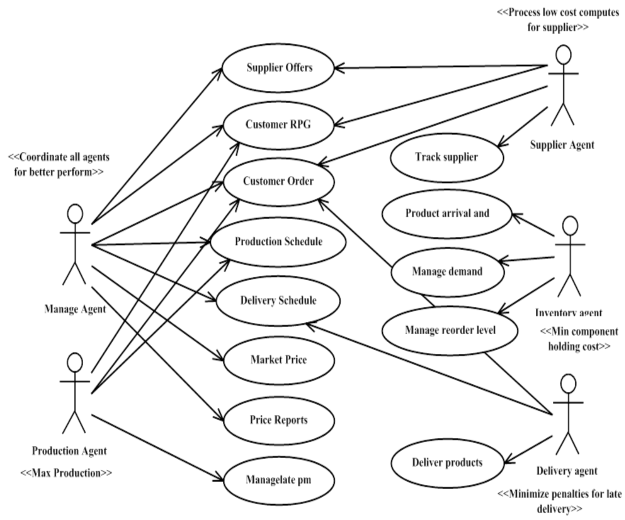

Fig.3. Use Case Diagram

5. NUMERICAL SIMULATION RESULTS FOR COOPERATIVE INDEX

We have evaluated our algorithm for static as well as dynamic model. We implemented the proposed algorithm and the results are presented. We have taken the perception of the author [25] which motivated us to coin the term CI. For illustration we have considered 5 agents in a software application. In these models we have quantified the cooperation among agents. This will be helpful to design and define the effective coordination among agents in a wide variety of environments.

5.1 STATIC MODEL

The figures presents the number of messages generated by the agents for accomplishing its own task (G_a) or the number of messages forwarded (W_a) by the agent based on the other agents' requests. It is observed in Fig.5 Agent3 (a_3) and Agent1 (a_1) has forwarded more messages compared to that of other agents. We can ensure that these agents are highly cooperative and so the utilization of the resources is also high. It is also interesting to infer that self processing of a number of messages from these agents is less. Hence if agents are generating more messages for self processing then they may not be at the required level of cooperation. The Fig.6 represents the number of messages each agent generated for the self processing. In other cases they may be highly cooperative. This inference is clearly reflected in Fig.7 with Agent1 (a_1) and Agent3 (a_3) having highest cooperative index out of the five agents we have considered.

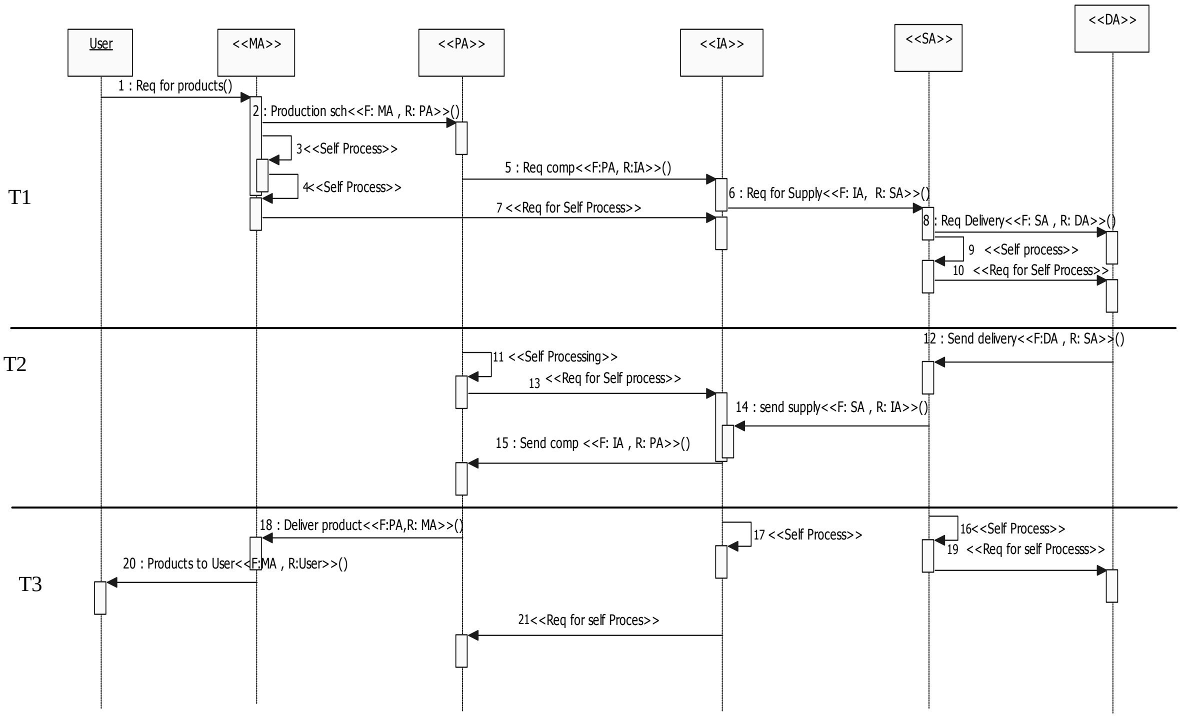

Fig.4. Sequence Diagram

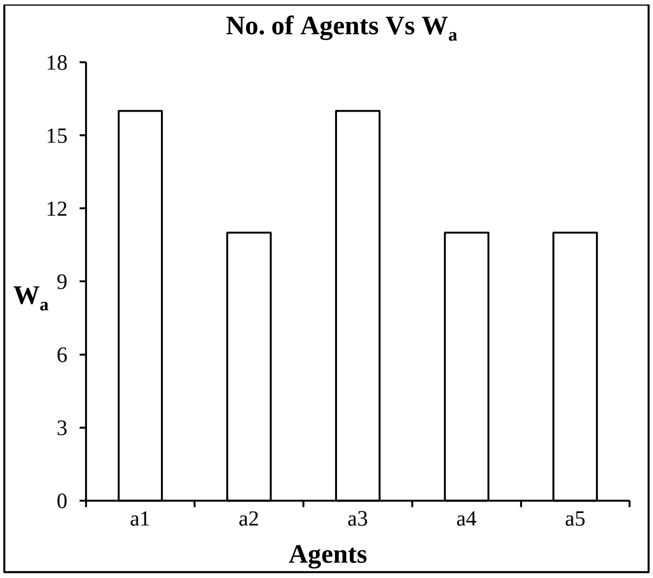

Fig.5. Agents Vs messages forwarded to other agents

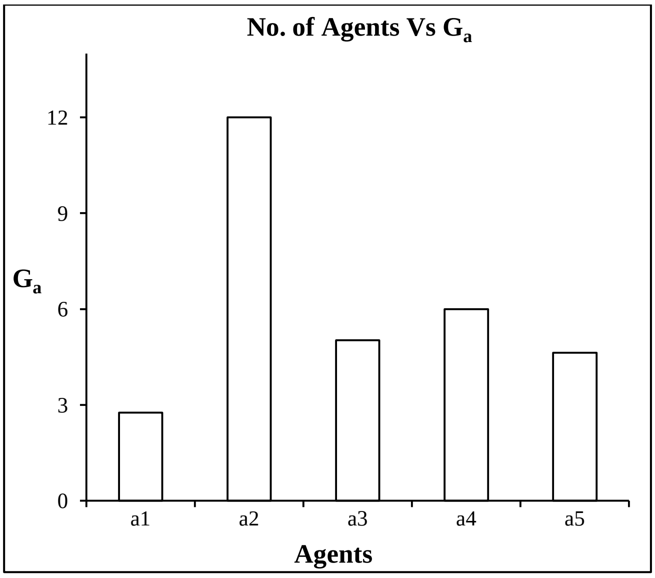

Fig.6. Agents Vs messages forwarded for self work

The W_a and G_a speak about messages generated, self processed, forwarded, sent without considering the resources across the network. Hence the calculation may not reflect a true cooperation among agents. Because of this there is a need to quantify cooperation among agents by considering resources in the network. It is also required to share resources in the execution environment. Hence we define a quantitative index called "Cooperative Index" by considering two different scenarios of the models as Static model and Dynamic model. CI essentially refers the level of cooperation between agents. We have used a range of values for R_u from 0 to 1. When R_u = 0 it is well understood that the CI value is equal to G_a, which means the agents may be in a position to share the resources. In our context this means agents do not have messages to be forwarded to other agents. When R_u = 1, the value of CI value is W_a and the agent behave in a cooperative manner to forward messages. It is important in a real situation to discuss the values when 0 < R_u < 1 the agents' behavior and we can review the execution environment based on CI. For example at R_u <= 5 resource usage or CI among agents is almost same. Thus CI helps in deciding execution environment. It is also important to note that the index considers both static and time dependent values while evaluating the level of cooperation.

5.2 DYNAMIC MODEL

We have considered four different time intervals for our discussion. When we are considering the time intervals we have taken into account the number forwarded, received and send during the previous time frames also. The Fig.8 to Fig.13 represents the numerical output with considering the time intervals. The Fig.8 represents the graphical information of time intervals T1, T2, T3, T4 and the agents' forwarded messages. The following figures clearly give an idea of the behavior of the different agents at each time interval. From Fig.8 at time T1 the Agent5 has forwarded more messages compared to other agents, at time T2 agent2 has forwarded more messages, at time T3 agent1 has forwarded more messages and at time T4 again agent1 has forwarded more messages. The Fig.9 represents the time intervals T1, T2, T3, T4 and the messages agents generated for self processing.

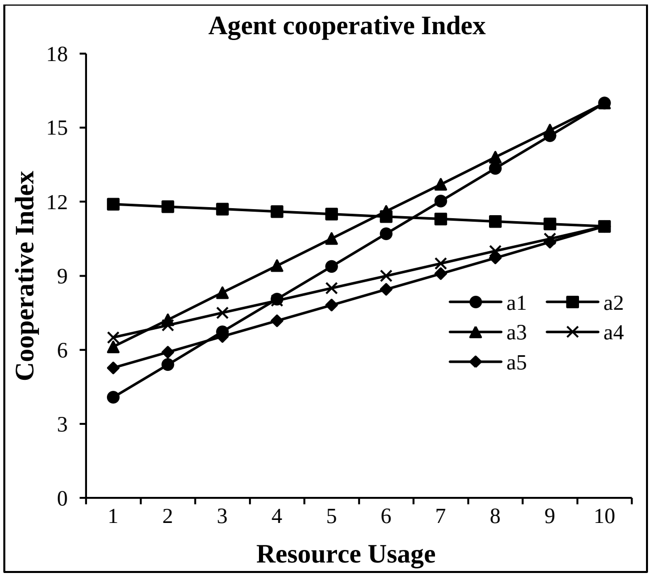

Fig.7. Resource usages versus cooperation Index

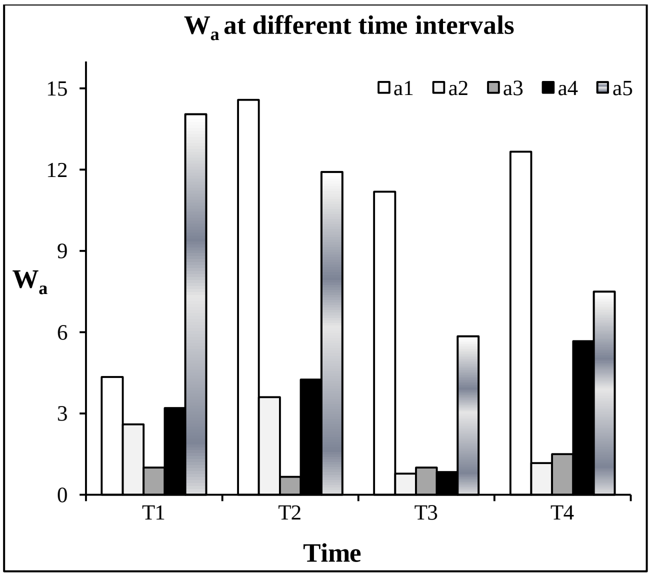

Fig.8. Number of Agents versus cooperation at different time intervals

In Fig.10 to Fig.13 represents the behavior of the different agent at each time interval T1, T2, T3, T4 against resource utilization. The Fig.10 represents the cooperative index of the five different agents at time T1. The output clearly represents that agent5 is highly cooperative. The Fig.11 represents the cooperative index of the five different agents at time T2, Fig.12 represents the cooperative index of the five different agents at time T3, and Fig.13 represents the cooperative index of the five different agents at time T4. The output in Fig.10 shows Agent5 is highly cooperative. The outputs from Fig.11, Fig.12, Fig.13 clearly represents that agent1 is highly cooperative. Thus by using this mathematical approach we can quantify the cooperation between the agents and the cooperation workload can be used with the representative work load for predicting the performance of agents in achieving the goals.

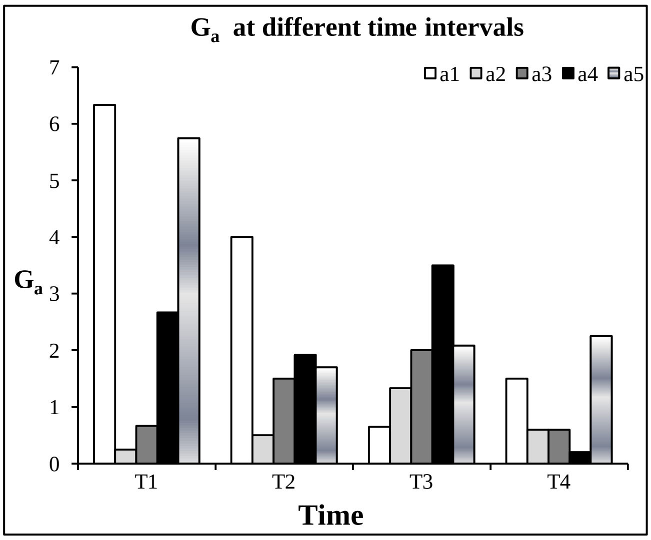

Fig.9. Number of Agents versus self process at different time intervals

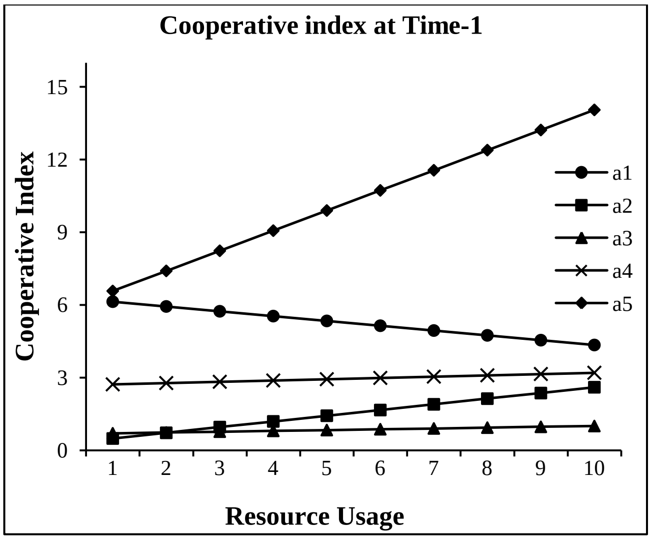

Fig.10. Resource Usage versus CI at time T1

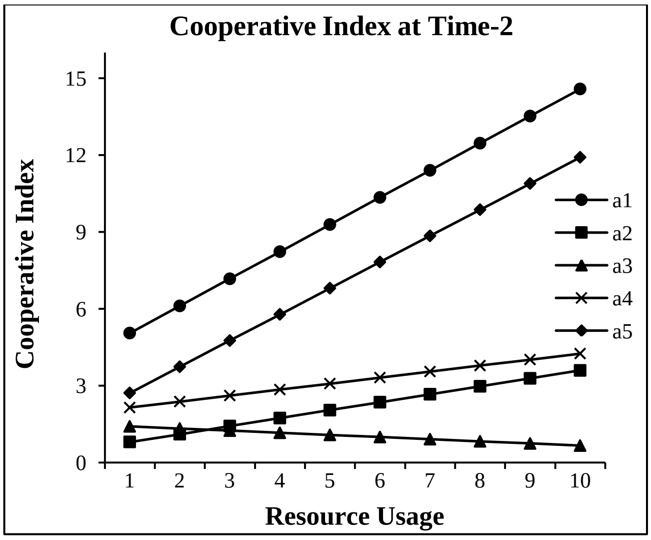

Fig.11. Resource Usage versus CI at time T2

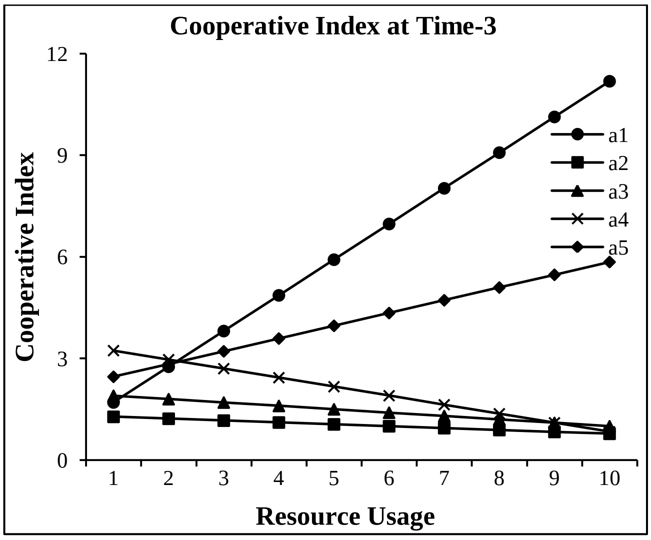

Fig.12. Resource Usage versus CI at time T3

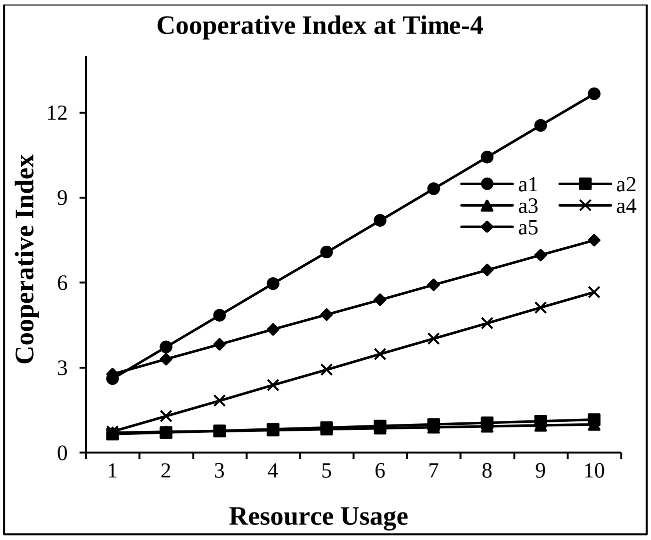

Fig.13. Resource Usage versus CI at time T4

6. REGRESSION ANALYSIS

We used regression analysis to ascertain the causal effect of the variable (W_a), (G_a) on the variable (CI) and the effect of the variable (R_u) on the effect of the variable (CI). In the regression study; we formulated the hypothesis about the relationship between the variables of interest, as cooperative nature (W_a), selfish nature (G_a) of the agent and resource sharing with cooperative index values obtained. The obtained results are tabulated below; from the results it is observed that cooperative index value is highly dependent on the variable W_a and R_u. This implies that if the agents are highly cooperative in nature then sharing of resources is more by those agents.

Table.1. Regression Analysis Data

Dependent Variable	Independent Variable	Correlation Value
W_a	CI	1 (strong)
G_a	CI	0.61538 (weak)
R_u	CI	1(strong)

7. CONCLUSION AND FUTURE WORK

Cooperation is one of the important fields of research in MAS. In this paper we proposed a novel approach for predicting the CI of agents in MAS. We represented cooperation between agents in the form of a mathematical model. The static and dynamic scenarios of the system are modeled explicitly by considering different situations and solved numerically. An algorithm is proposed which describes a UML model for identifying the number of agents; the messages and a mathematical model for assessing the level of cooperation.

There are several other aspects that we want to explore further. The non-functional requirements such as performance, scalability, reliability, maintainability, adaptability plays an important role in a real time system. To improve the efficiency of the MAS system we have to address the issues such as response time, throughput, device utilization etc. We propose to address the performance issues of MAS and a suitable deployment environment depending on the CI of agents in the early stages of the SDLC.

REFERENCES

[1] Gerhard Weiss, "*Multi Agent Systems: A Modern Approach to Distributed Artificial Intelligence*", MIT press, 2000.

[2] P. Maes, "Agents that reduce work and information overload", *Communications of the ACM*, Vol. 37, No. 7, pp. 30 - 40, 2000.

[3] M. Ljunberg and A. Jucas, "The OASIS air traffic management system", *Proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence*, Vol. 2, pp. 1003, 1992.

[4] J. E. Doran, S. Franlinn, N. R. Jennings and T. J. Norman, "On Cooperation in Multi-agent Systems", *The Knowledge Engineering Review*, Vol. 12, No. 3, pp. 309 - 314, 1997.

[5] A. Ebrahim, "A Conceptual Agent Cooperation Model for Multi-Agent Systems' Team Formation Process", *Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology*, Vol. 1, pp. 12 - 20, 2008.

[6] Y. T. Kotb, S. S. Beauchemin and J. L. Barron, "Petri Net-Based Cooperation In Multi-Agent Systems", *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, pp.123 - 130, 2007.

[7] Michael Wooldridge and Nicholas R. Jennings, "The Cooperative Problem-Solving Process", *Journal of Logic and Computation*, Vol. 9, No. 4, pp. 563 - 592, 1999.

[8] E. Semsar-Kazerooni and K. Khorasani, "A Game Theory Approach to Multi-Agent Team Cooperation", *American Control Conference*, pp. 4512 - 4518, 2009.

[9] Abdellah Bedrouni, Ranjeev Mittu, A. Boukhtouta and J. Berger, "*Distributed Intelligent Systems*", Springer, 2009.

[10] Giovanni Caire et al., "Agent oriented analysis using Message/UML", *Proceedings of the 2nd International Workshop on Agent-Oriented Software Engineering II*, pp.119 - 135, 2001.

[11] F. Giunchiglia, J. Mylopoulos and A. Perini, "The Tropos Software Development Methodology: Processes, Models and Diagrams", *Proceedings of the First International Conference on Autonomous Agents & Multi-Agent Systems*, pp. 35 - 36, 2002.

[12] Joseph A. Giampapa and Katia P. Sycara, "Team-oriented Agent Coordination in the RETSINA Multi-Agent System", Technical Report CMU-R1-TR-02-34, The Robotics Institute, Carneige Mellon University, *Presented at AAMAS Workshop on Teamwork and Coalition Formation*, 2002.

[13] Katia Sycara and Michael Lewis, "From Data to Actionable Knowledge and Decision", *Proceedings of the Fifth International Conference on Information Fusion*, Vol. 1, pp. 577 – 584, 2002.

[14] Nicholas R. Jennings, Katia Sycara and Michael Wooldridge, "A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-agent Systems*, Vol. 1, No. 1, pp. 7 - 38, 1998.

[15] D. I. Mark, "Cooperation Structures", *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 600 - 605, 1997.

[16] Ping Wang, "A Brief Survey on Cooperation in Multi-Agent System", *Proceedings of the International Conference on Computer Design And Applications*, Vol. 2, pp. V2-39 – V2-43, 2010.

[17] R. G. Smith and R. Davis, "Frameworks for Cooperation in Distributed Problem Solving", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 11, No. 1, pp. 61 - 70, 1981.

[18] Roie Zivan, Alon Grubshtein Michal Friedman and Amnon Meisels, "Partial Cooperation in Multi-Agent Search", *Proceedings of the 11th International Conference on Autonomous Agents and Multi-agent Systems*, Vol. 3, pp. 1267 - 1268, 2012.

[19] O. D. Reza, F. J. Lex and M. M. Richard, "Consensus and Cooperation in Networked Multi-Agent Systems", *Proceediings of the IEEE*, Vol. 95, No.1, pp. 215 - 233, 2010.

[20] G. Adel, R. M. Habib and M. Reza, "A Novel Algorithm for Coalition Formation in Multi-Agent Systems using Cooperative Game Theory", *Proceedings of the 18th Iranian Conference on Electrical Engineering*, pp. 512 - 516, 2010.

[21] R. L. Victor, "Cooperative Multiagent Systems: A Personal View of the State of the Art", *IEEE Transactions on Kowledge and Data Engineering*, Vol. 11, No. 1, pp. 133 - 142, 1999.

[22] C. Gutierrez and M. I. Garcia, "A Metric Suite for the Communication of Multi-agent Systems", *Journal of Physical Agents*, Vol. 3, No. 2, pp. 7 - 15, 2009.

[23] G. Wojciech and K. Halina, "Coalition Formation in Multi-agent Systems - an Evolutionary Approach", *Proceedings of the Multiconference on Computer Science and Information Technology*, pp. 125 – 130, 2008.

[24] Krishna Kavi, David C. Kung, Hitesh Bhaambhani, Gaurav Pancholi and Marie Kanikarla, "Extending UML to Modeling and Design of Multi-Agent Systems", *Proceedings of workshop on Software Engineering for Large Multi-Agent Systems*, 2003.

[25] R. L. Victor, "Reflections on the Nature of Multi-Agent Coordination and its Implications for an Agent Archicture", *Autonomous Agents and Multi-Agent Systems*, Vol. 1, pp. 89 - 111, 1998.

[26] L. Buttyan and J. P. Hubaux, "Stimulating Cooperation in Self-organizing Mobile ad-hoc Networks", *ACM/Kluwer Mobile Networks and Applications*, Vol. 8, No. 5, pp. 579 - 592, 2003.

[27] A. Urpi, M. Bonucceli and S. Giordano, "Modelling Cooperation in Mobile Adhoc Networks: A Formal Description of Selfishness", *Proceedings of Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 3 - 5, 2003.

[28] T. Moyaux, B. Chaib-draa and S. D'Amours, "*Multiagent based Supply Chain Management*", Springer, 2006.