Framework for Multi-Agent Systems Performance Prediction Process Model: MASP^3

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

Submitted in August, 2013; Accepted in September, 2014

Abstract - Multi Agent System is one of the upcoming areas in the research/industry for building complex distributed application. MAS encompass multiple features for distributed application. For complex applications the non-functional requirements has the same importance as functional requirements. The non-functional requirements are performance, reliability, maintainability etc. We are focusing on the prediction of performance by considering the characteristics of agents in the early stages of MAS development. Evaluation of the performance at the end of software development leads to increase in the cost of design change. To compare design alternatives or to identify system bottlenecks, the quantitative system analysis must be carried out from the early stages of the software development life cycle. In this paper we propose a framework for predicting performance of a Multi-Agent System by considering the characteristics of agents in the early stages of software development.

Index Terms – Multi-Agent System, Performance Prediction Process Model, UML, Agent Characteristics, Software Performance Engineering.

1.0 INTRODUCTION

Designing and building high quality industrial-strength software is difficult. Indeed, it has been claimed that such development projects are among the most complex construction tasks undertaken by humans. Each successive development either claims to make the engineering process easier or to extend the complexity of applications that can feasibly be built. Although there is some evidence to support these claims, researchers continually strive for more efficient and powerful software engineering techniques, especially as solutions for ever more demanding applications are required. There are compelling arguments for believing that an agent oriented approach will be of benefit for engineering certain complex software systems. These arguments have evolved from a decade of experience in using agent technology to construct large-scale, real world applications in a wide variety of industrial and commercial domains [1]. When adopting an agent oriented view of the world, it soon becomes apparent that a single agent is insufficient. Most problems require or involve multiple agents: to represent the decentralized nature of the problem, the multiple loci of control, the multiple perspectives, or the competing interests. Some of the important characteristics of the agents which distinguish an agent from objects are Autonomous, Cooperation, Goal oriented, Adaptability, Mobility, Negotiation etc. Analyzing, designing and implementing software as a collection of interacting, autonomous agents i.e., as a multi-agent system [2,3] represent a promising point of departure for software engineering. Whatever the complexity of the system the quality of the system cannot be neglected. The important non functional characteristics of the systems are performance, reliability, availability, maintainability etc.

Performance is an important but often neglected aspect of software development methodologies. Performance refers to system responsiveness, either the time required to respond to specific events, or number of events processed in a given time interval. Performance problems may be so severe that they require extensive changes to the system architecture. If these changes are made late in the development process, they can increase development costs, delay deployment, or adversely affects other desirable qualities of a design, such as understandability, maintainability, or reusability. Finally, designing for performance from the beginning produces better systems than using a 'fix-it-later' approach. Software Performance Engineering (SPE) has evolved over the past years and has been demonstrated to be effective during the development of many large systems [4]. Although the need for SPE is generally recognized by the industry, there is still a gap between the software development and the performance analysis domains. In this paper we propose a framework for predicting the performance of MAS using SPE techniques by considering the characteristics of agents in the MAS.

2.0 RELATED WORK

Agent-Oriented Software Engineering (AOSE) is being described as a new paradigm for the research field of Software Engineering. Some of the very widely used AOSE methodologies are MESAGE, MasE, Gaia, Tropos [5]. In [6] the author discusses the importance of performance engineering in Agent systems and suggested to define benchmarks and metrics that help to compare and contrast different Agent systems to support software engineering themes within Agent systems. In [7] estimating costs for agent oriented software is discussed. The scalability issue in Multi Agent System (MAS) is addressed in [8]. The main objective of this author was to combine performance engineering with agent oriented design methodologies to design and build large agent based applications. The author presents a solution for performance engineering of mobile agent systems during the development of agent code. In [9] a novel approach for performance improvement of Multi-Agent based system architecture is discussed. The authors also proposed a metrics suit for evaluating agent-oriented architectures. Most of the metrics are

[1, 2, 3] *M. S. Ramaiah Institute of Technology, Bangalore*
E-mail: [1]*ajithasankar@gmail.com*

inspired by the object-oriented metrics but they are adapted to agent oriented concepts. Williams and Smith in [10] applied the SPE methodology to evaluate the performance characteristics of a software architecture specified by using the Unified Modeling Language (UML) diagrams, ITU Message Sequence Chart (MSC) features. The extension to the SPE process and its associated models for assessing object oriented distributed systems are described in [11]. The use of SPE for web applications during software architectural design phase is discussed in [12]. An extension of the SPE approach was developed by Cortellessa and Mirandola in [13]. The proposed methodology, called PRIMA-UML, makes use of information from different UML diagrams to incrementally generate a performance model representing the specified system. The technique considered OMT based Class diagrams, Interaction diagrams and State Transition diagrams to specify the software systems and defined an intermediate model, called Actor-Event Graph, between the specification and the performance model. SAP-ONE [14] is a methodology for the performance modeling of a software system. It consists essentially in the annotated generation of a queuing network from an UML architectural model. Even though several of these approaches have been successfully applied, still there is a gap in integrating the software performance prediction process into the SLDC. We propose a framework based on the paper [15].

3.0 SOME CRITICAL ISSUES

- In general the software performance prediction starts from the analysis phase and continues throughout the SDLC. Most of the researches in MAS are concentrated on the AOSE methodologies, and implementation methodologies, agent theories, architectures and tools. The SPE approach for MAS is not addressed in the literature.
- Accessing the performance of MAS during the feasibility study of MAS development.
- Performance of the agents by considering the different characteristics (cooperation, negotiation, mobility, etc)
- Performance issues such as load balancing, scheduling and resource allocation for MAS in the context of SPE can be addressed.

4.0 PROSPOSE METHODOLOGY

The proposed frame work for predicting the performance of MAS in the early stages of software development is expressed in the form of flowchart in Figure1. The activities involved in the elements of the process model are:

1. **Consider the Agent Characteristics:** MAS are formed by the coalition of more than one agent. The characteristics of agents are autonomous, cooperation, negotiation, mobility etc which distinguishes the agents from objects. These characteristics of agents have a high influence on the performance of the system. So consider the characteristics of agents and define the SPE assessments for the application considered. In our work we considered the characteristics cooperation and negotiation of agents.
2. **Define the SPE assessments for a given software application:** SPE is applied to predict the performance of software systems early in the development life cycle. In our framework the possibility of assessing the performance during the feasibility study of agents by considering the characteristics of the agent is addressed. We have considered the SPE assessments such as acceptable response time or constraints on resource requirements based on the approach on [10].
3. **Develop UML models augmented with performance parameters:** Capture the performance requirement data by modeling the application using a modeling technique. Since UML is a universally accepted modeling language, we have extended the UML to model the application developed using agents
4. **If the transformation technique is necessary to get the performance models, apply the transformation technique and derive the performance models from UML models:** Use traditional transformation algorithms, e.g graph grammar techniques, XSLT, to generate the performance model from UML models. In our work we have devised an algorithm to transform a UML sequence diagram into agent execution graph along with the demand vector by considering the agent character “Cooperation”.
5. **Alternatively, generate Performance Models from UML models:** Generate performance models from UML models by mapping the elements. We have proposed to map the elements of the UML models to the ANN model.
6. **Solve the model:** The models can be solved analytically or by simulation. Simple systems can be solved analytically, whereas simulation is preferred for larger and complex systems. So we propose to devise algorithms for solving the models analytically as well as by simulation.
7. **Report the performance metrics,** If the performance metrics obtained by considering one of the agent characteristic is acceptable, then proceed the same cycle for the next character of the agent
8. **Alternatively,** if the performance metric obtained for the particular characteristic is acceptable then the same methodology can be extended to the other characteristics.

5.0 CASE STUDY

Agent technology is adopted by different areas of applications [16,17]. The case study we have considered to validate our methodology is a sub module of supply chain management system [18]. The module we have considered consists of three agents namely Production Agent, Supply agent and Delivery agent. The use case diagram is used to represent a high level abstraction of the system. The use case diagram consists of the agents and the use cases. From this diagram we can identify the number of agents in the system and the interaction of the agents with the use cases. The use case diagram is presented in Figure2. The model is simulated using the tool SMTQA(Simulation of Multi Tier Queing Applications) for 1000 requests[19]. We have considered MAS with 3 agents. The inputs required for simulation are: software resource requirements, execution environment, software execution structure, and resource usage. The probability of occurrence of the use cases is considered from the cooperative index of the agents. The size of the use cases is estimated using the use case point approach. We have assumed the processing speed of the server as 2000 KB/sec and the speed of the internet considered is 96.8 KB/sec. The mean arrival rate we considered is 0.05. The performance metrics for the agents in the MAS are obtained and tabulated in the Table1.

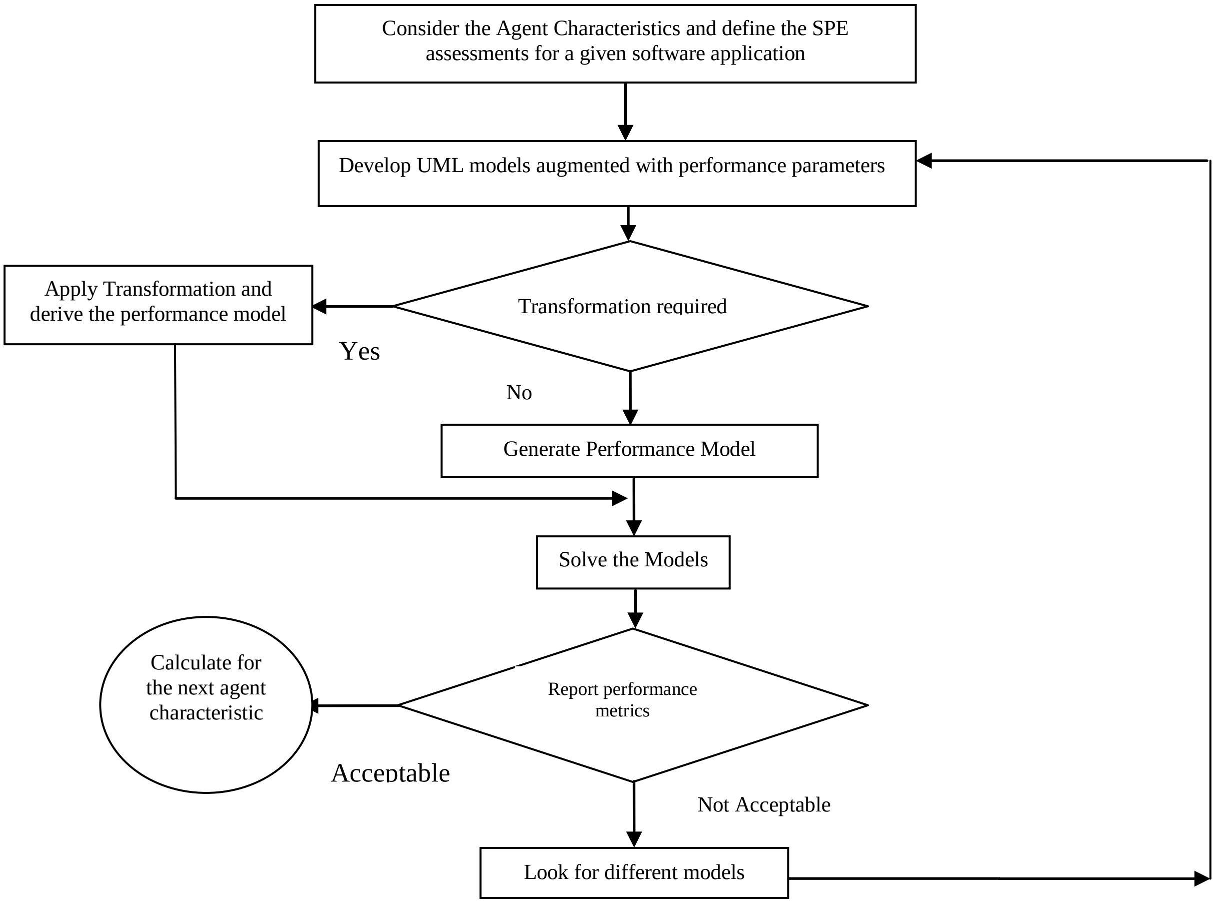
Consider the Agent Characteristics and define the SPE assessments for a given software application
Develop UML models augmented with performance parameters
Transformation required
Apply Transformation and derive the performance model
Yes
No
Generate Performance Model
Solve the Models
Report performance metrics
Calculate for the next agent characteristic
Acceptable
Not Acceptable
Look for different models

Figure1: Framework for Performance Prediction Process Model

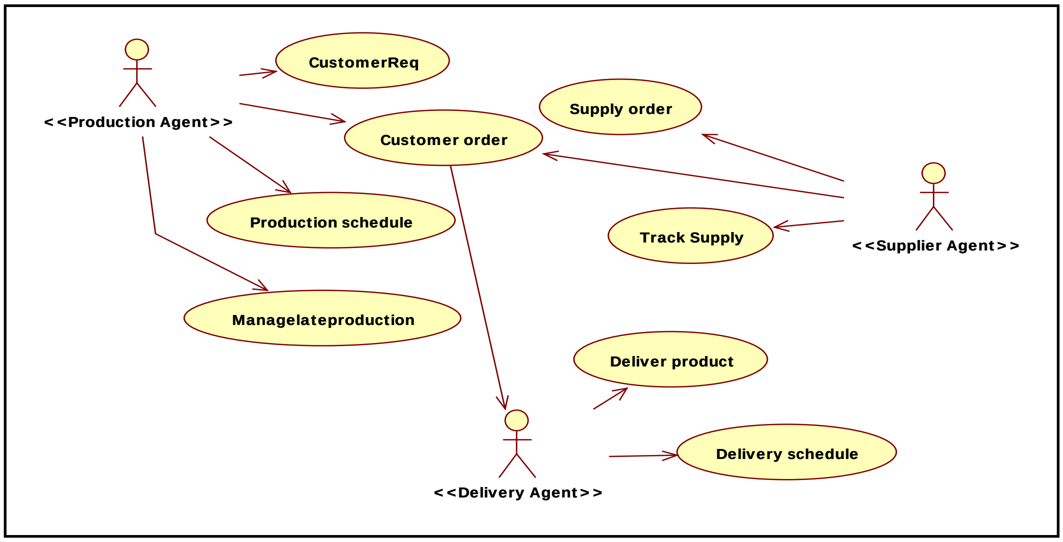

Figure 2: Use Case diagram for case study

	Average Response time (Sec)	Average Service Time(Sec)	Average Waiting Time(Sec)	Probability of Idle Server(Sec)	Probability of Dropping of Sessions(Sec)
Production Agent	0.417	0.040	0.377	0.0700	0.627
Supply Agent	0.023	0.019	0.004	0.724	0.000
Delivery Agent	0.020	0.017	0.003	0.771	0.000
Internet	0.014	0.011	0.003	0.621	0.017

Table 1: Performance Metrics obtained for MAS using SMTQA

6.0 CONCLUSION

This paper discusses a framework for MASP3 model that allows modeling MAS with the goal of assessing performance of the system during the feasibility study and early in the Software Development Life Cycle (SDLC). This framework describes the elements of MASP3 model and provides flexibility to integrate the software performance prediction process with SE process. The proposed MASP3 Model provides the possibility of deriving performance models from Unified Modeling Language (UML) models by mapping the UML model elements into the elements of multitier architecture simulation model and solving the simulation model. The description and significance of each element in the process are described in this paper. We considered a case study in supply chain management system to validate the framework.

7.0 REFERENCES

[1]. N. R. Jennings and M. Wooldridge (eds,) (1998) “Agent technology: foundations, applications and markets” Springer Verlag.

[2]. G.M.P. O'Hare and N. R. Jennings (editors) (1996) “Foundations of distributed artificial intelligence” John Wiley & Sons.

[3]. Gomaa H, Menasce D.A: “Design and Performance Modeling of Component Interconnection Patterns for Distributed Software Architectures, Proceedings of the WOSP 2000, ACM Second International Workshop on Software and Performance, Ottawa, Canada, September 2000.

[4]. Connie U. Smith, Performance Engineering of Software Systems, Reading, MA Addison-Wesley, 1990.

[5]. Lind J. Issues in Agent-oriented Software Engineering. The first International workshop on Agent oriented Software Engineering (AOSE-2000).

[6]. Omer Rana, Chris Preist, Michael Luck: “Progress in Multi-Agent Systems Research, March 2000

[7]. Jorge J. Gomez-Sanz, Juan Pavon, Francisco Garito: “Estimating Costs for Agent Oriented Software”, TIC2002-04516-C03-03

[8]. Elema Gomez-Martinez Sergio Ilarrai Jose Merseguer “Performance Analysis if Mobile Agent tracking, WOSP '07, Feb G. Schneider and J. P. Winters: “Applying Use Cases, SecondEdition”, Addison Wesley (2001).

[9]. Ivan Garcia-Magarino, Maximo Cossentino, Valeriascidita “A metric suit for evaluating agent oriented architecture”, SAC'10-Mar 22/26.2010, Sierre, Switzerland ACM978-1-60558-638-0/10/03

[10]. C.U.Smith and Lioyd G.Williams., Performance Engineering Models of CORBA-based distributed-object systems. s.l. : Performance Engineering Services and Software Engineering Research, 1998.

[11]. Peter Utton, et al., Performance Analysis of Object-oriented Designs for Distributed Systems. University of Kent at Canterburry : Technical Report, 1999

[12]. Connie U.Smith and Lioyd G. Williams., "Building Responsive and Scalable Web Applications." December 2000. Proceedings CMGC

[13]. V. Cortellessa and R.Mirandola, "Deriving a Queueing Network Based Performance Model from UML Diagrams." s.l. : ACM Proc. intl, 2000. Workshop Software and Performance. pp. 58-70.

[14]. A. Di Marco, P. Inverardi “Compositional generation of Software Architecture Performance QN Models”. Proc. 4th Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004.

[15]. D.Evangelin Geetha, Dr.T.V.Suresh Kumar, Dr.K.Rajanikanth,”Framework for Hybrid Performance Predition Process Model: Use Case Engineering Approach”ACM Sigsoft Software Engineeering Notes, May 2011,Volume 36 Number 3.

[16]. Tomasek.M, Trelova. J, "An e-commerce applications based on the multi-agent system," Emerging eLearning Technologies & Applications (ICETA), 2012 IEEE 10th

International Conference on vol., no., pp.391,394, 8-9 Nov. 2012, Stara Lens

[17]. Islam, S. R.; Sutanto, D.; Muttaqi, K.M., "Application of multi-agent system for preventing power interruption in a large power system," Harmonics and Quality of Power (ICHQP), 2012 IEEE 15th International Conference.

[18]. T.Moyaux, B.Chaib-draa and S.D'Amours. Multiagent based Supply Chain Management, chapter Supply Chain Management and multiagent Systems, Springer-Verlag Berlin Heidelberg, 2006.

[19]. D.Evangelin Geetha, T.V. Suresh Kumar, Performance Modeling and evaluation of Distributed Systems, Ph.D thesis, Visvesvaraiah Technological University, Karnataka, 2012.