

Modeling the Negotiation of Agents in MAS and Predicting the Performance – an SPE Approach

S. Ajitha

Ramaiah Institute of Bangalore-54, India

E-mail: ajithasankar@gmail.com

Keywords: MAS, negotiation, conditional probability, SPE, work load

Received: July 15, 2019

Software performance engineering (SPE) process starts at the early stages of software development life cycle which helps to develop software that meets the performance requirements on time and budget. Multi-Agent Systems (MAS) are comprised of one or more agents who coordinate each other to accomplish some task. The coordination can be achieved through cooperation and negotiation. In the early development stages measuring the negotiation workload and predicting the performance remains an important but largely unsolved problem. The problem of uncertainty regarding the negotiation workload is required to be addressed by estimation techniques. Hence, in this research we developed a probabilistic model for the negotiation scenario among the agents in a given time horizon. The negotiation workload obtained from the probabilistic model is integrated with the representative workload of the agents for predicting the performance of agents in MAS. The tool SMTQA is used for obtaining the performance metrics. Analysis of the execution environment is done by considering various configurations in the hardware resources based on the dynamic workload of the negotiation agents over a time horizon. From the sensitivity analysis, the bottleneck resources are identified and suggestions for improvement are proposed.

Povzetek: Predstavljena je izvirna metoda za ocenjevanje in napovedovanje delovanja večagentnih sistemov.

1 Introduction

A Multi-Agent System (MAS) is usually understood as a system composed of interacting autonomous agents. MAS have been employed successfully in a number of scenarios. The important characteristics of the agents which distinguish it from an object are Autonomous, Cooperation, Goal oriented, Adaptability, Mobility, Negotiation etc. Many articles on MAS have been mainly concerned with functional characteristics such as coordination, rationality and knowledge modeling. The nonfunctional characteristics also have the equal importance as the functional characteristics for any software system [1-5].

This research aims at making a contribution towards the non-functional characteristics Performance of agents in a MAS by considering the negotiation character of agents. Software Performance Engineering (SPE) is a method for constructing software systems to meet the performance objectives at the early stages of Software Development Life Cycle (SDLC). In SPE, system does not exist so it is not possible to develop the work load parameters from measurement data. Therefore, models of the system are used to collect the data required to predict the performance. The different data required for the SPE approach are Workload scenarios, Performance goals, Software Design concepts, Execution environment and Resource usage estimates [5-7].

The performance prediction of agents by considering the cooperation character of agent the authors have

published different articles in [8-11]. In the early development stages measuring the negotiation workload and predicting the performance remains an important but largely unsolved problem. The problem of uncertainty regarding the negotiation workload is required to be addressed by estimation techniques. Hence, in this chapter a model for the negotiation scenario among the agents in a given time horizon is developed. The fitness function which represents the fitness of the agent in the negotiation scenario among 'n' agents in MAS is considered while framing the model. The negotiation workload obtained from the probabilistic model is integrated with the representative workload of the agents for predicting the performance of agents in MAS [12-14].

The tool SMTQA is used for obtaining the performance metrics. The execution environment is analyzed by considering various configurations in the hardware resources based on the dynamic workload of the negotiation agents over a time horizon. From the sensitivity analysis, the bottleneck resources are identified and suggestions for improvement of the software are made [15].

2 Methodology

A methodology is proposed to model the negotiation scenario of agents in a MAS using probabilistic approach. This is based on the methodology discussed for the

distributed systems in [15]. Let $t \in [t_0, T + t_0 - 1]$ be the interval, where T be the number of intervals, t_0 be the initial interval of the given time horizon. Let S_T be the negotiation services that are considered to be executed during the T intervals. With these assumptions, we have devised the methodology as follows,

- Developing a mathematical model for demand of negotiation services over a given time horizon
- Modeling the resources in the execution environment
- Modeling the variations (alternate designs) in the execution environment
- Identifying bottleneck resources and improving the performance by sensitivity analysis

2.1 Modeling of demand of negotiation services

Consider a MAS ‘A’ with ‘n’ number of agents. Let S_T be a set of negotiation services that to be executed during the T time intervals.

Let $S_T = \{S_1, S_2, S_3, \dots, S_m\}$ be the negotiation services.

Let $W_1, W_2, W_3, \dots, W_m$ be the size (representative workload) of the negotiation services.

Let $P_{ijk}(t)$ be the probability that agent ‘i’ communicates with the agent ‘j’ with the work load of W_k at the time interval t. The sum of these $P_{ijk}(t)$ over k equals 1.

Each negotiation service that can be occurred in the interval $t \in [t_0, T + t_0 - 1]$, is characterized by:

- $P_{ijk(t),e}$ – probability of occurrence of the e^{th} negotiation primitive of those specified at time interval t
- $D_{i,e}^s$ – expected demand for each negotiation services $s \in S_T$, if the e^{th} primitive occurs among those specified at time interval t.

Based on such specification of expected primitives, s_t demand scenarios can be generated in each interval t, where $p_{ij,s}(t)$ be the probability that s^{th} scenario occurs at time t when agent ‘i’ negotiates with the agent ‘j’.

In the first period t_0 :

$$S_{t_0} = E_{t_0} \tag{1.1}$$

while in the following interval $t \in [t_0, T + t_0 - 1]$, the

number of scenarios can be recursively computed as:

$$S_t = E_t \cdot S_{t-1} \tag{1.2}$$

Each workload scenario can be defined as the occurrence of one event at period t given one scenario in the previous period t-1. Definition of the demand scenarios based on the specification of six events over a

time horizon constituted of three intervals is given in Table 1.

Period	Event	Probability	Demand
0	1	p_1	D_1
	2	p_2	D_2
1	3	p_3	D_3
	4	p_4	D_4
2	5	p_5	D_5
	6	p_6	D_6

Table 1: Definition of the demand scenarios.

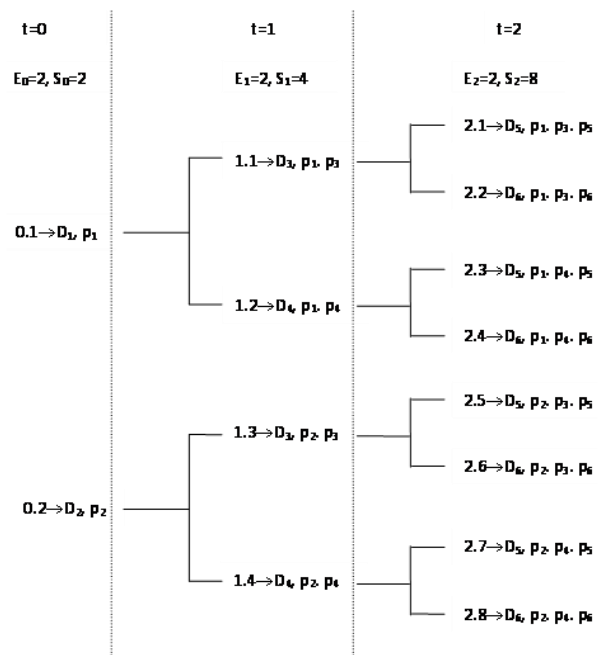


Figure 1: Conditional probability tree for the workload scenario.

2.2 Calculation of workload

The model is simulated by considering three time intervals such that for a given time interval t, $t \in [0, 2]$. The state (negotiation scenario) of the application in time t depends on the state (negotiation service scenario) at the time interval t-1 and the type of the request arrived at t. Hence the scenarios of the negotiation service are considered as states of the software application and the pattern of execution of negotiation services are modeled using the UML, State Chart Diagram. The Figure 2 to Figure 4 represent the workload to be executed during the different time duration. The negotiation services which are having a very less workload are executed during time t=0. During the time t=1 the negotiation services having a higher workload are executed. At time t=2, the negotiation services having an average workload are executed. From Figure 5, it is observed that agent a1 and agent a3 are negotiating more with agent a4. Also agent a2, agent a4, agent a5 are negotiating more with agent a3.

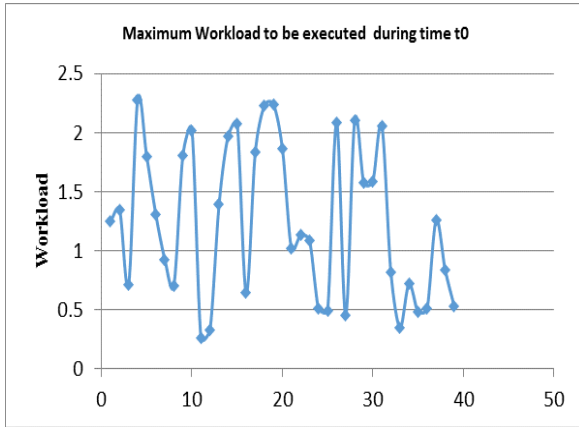


Figure 3: Work Load at Time t0.

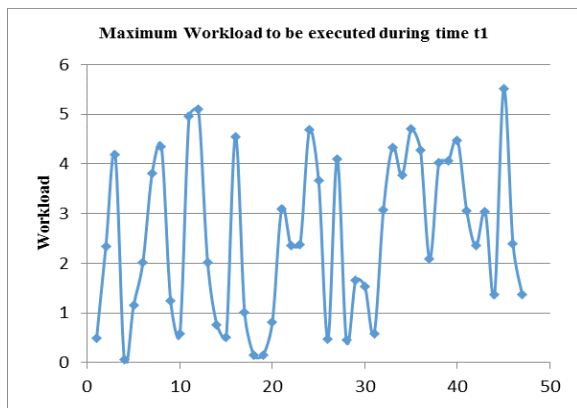


Figure 4: Work Load at Time t1.

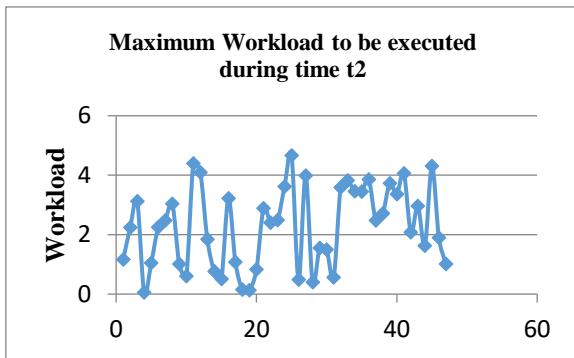


Figure 5: Work Load at Time t2.

2.3 Simulation results

The scenarios of the negotiation primitives are simulated using the tool SMTQA, and the performance metrics are obtained and tabulated in Table 2. The columns in the table represents average response time (Avg Resp Time), average service time (Avg Serv Time), average waiting time (Avg Wait Time), probability of idle time (Prob Idle) and average dropping of requests (Avg Drop). The rows represent the five agents and the internet. (IntN). From the values it is observed that the number of negotiation services dropped is high in Agent 2 and in the Internet. Hence these two resources are identified as the bottleneck resources. To solve this problem, we conducted the

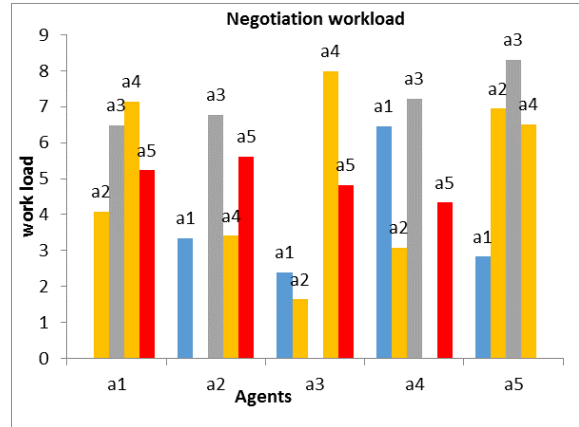


Figure 2: Agents V/S Workloads.

	Avg Resp Time	Avg Serv Time	Avg Wait Time	Prob Idle	Avg Drop
Agt 1	0.003	0.003	0.0	0.342	0.0
Agt 2	0.607	0.021	0.586	0.047	0.830
Agt 3	0.006	0.006	0.0	0.196	0.0
Agt 4	0.008	0.007	0.001	0.181	0.004
Agt 5	0.016	0.013	0.002	0.218	0.005
IntN	0.060	0.021	0.586	0.047	0.830

Table 2: Simulation Result for the configuration C1.

sensitivity analysis by considering different configurations and are presented in the next section.

2.4 Sensitivity analysis

Simulation of the behavior of the resources is carried out by considering the configurations C1 to C6 as follows. C1:-Processing speed of CPU is 2000, and the Internet speed assumed is 96.

C2:-Processing speed of CPU is 3000, and the Internet speed assumed is 96.

C3:-Processing speed of CPU is 4000, and the Internet speed assumed is 96.

C4:-Processing speed of CPU is 2000, and the Internet speed assumed is 146.

C5:-Processing speed of CPU is 3000, and the Internet speed assumed is 146.

C6:-Processing speed of CPU is 4000, and the Internet speed assumed is 146.

The results of the different simulation runs are presented in the form of tables. The results obtained for Agent 1 for the different configurations considered is presented in Table 3. The maximum time taken by the Agent 1 to respond is 0.036 in the configuration C4 and minimum time taken to respond is 0.003 with configuration C1. The waiting time in Agent1 is also maximum for the configuration C4. This is due to the configuration of C1; the number of negotiation services dropped is more due to the low configuration of the

Internet. Hence the number of negotiation services that are processed by Agent 1 is less compared to other configurations. In configuration C4, the processing speed of the Internet is increased so that the negotiation services by Agent 1 are more.

Agent 1					
	Avg Res Time	Avg Serv Time	Avg Wait Time	Pro Idle	Avg drop
C1	0.003	0.003	0	0.342	0
C2	0.02	0.019	0.001	0.196	0
C3	0.013	0.012	0	0.132	0
C4	0.036	0.032	0.004	0.135	0
C5	0.017	0.017	0	0.271	0
C6	0.016	0.015	0.001	0.137	0.003

Table 3: Simulation results obtained for Agent 1 with different Configuration.

The results obtained for Agent 2 for the configurations considered are tabulated in the Table 4. Agent 2 has taken the maximum time 0.607 to respond under Configuration C1 and the minimum time to respond is 0.009 with configuration C5. Figure 21 presents the average dropping of requests and probability of idle time of Agent 2. The maximum number of requests is dropped in configuration C1. The maximum waiting time for the requests is observed with configuration C1. This has happened because Agent2 has received more requests.

Agent 2					
	Avg Res Time	Avg Serv Time	Avg Wait Time	Pro Idle	Avg drop
C1	0.607	0.021	0.586	0.047	0.83
C2	0.02	0.15	0.004	0.108	0.005
C3	0.01	0.009	0.001	0.123	0
C4	0.023	0.018	0.005	0.115	0.005
C5	0.009	0.008	0.001	0.159	0
C6	0.01	0.009	0.001	0.133	0

Table 4: Simulation results obtained for Agent 2 with different Configuration

The results obtained for Agent 3 for the configurations considered are presented in Table 9.5. Agent 3 has taken the maximum time 0.118 to respond under Configuration C4 and the minimum time to respond is 0.003 with configuration C5. The average dropping of requests and probability of idle time of the Agent 3 is plotted in Figure 23. The maximum number of requests is dropped in configuration C4. The maximum waiting time for the requests is observed in configuration C4 for Agent 3.

Agent 3					
	Avg Res Time	Avg Serv Time	Avg Wait Time	Pro Idle	Avg drop
C1	0.006	0.006	0	0.196	0
C2	0.008	0.008	0.001	0.14	0
C3	0.094	0.052	0.042	0.083	0.073
C4	0.118	0.06	0.059	0.082	0.102
C5	0.003	0.003	0	0.255	0
C6	0.055	0.033	0.022	0.099	0.044

Table 5 Simulation results obtained for Agent 3 with different Configuration.

The Table 6 presents the results obtained for Agent 4 for the different configurations considered Agent 4 has taken the maximum time 0.033 to respond under Configuration C3 and the minimum time to respond is 0.008 with configuration C1. The maximum number of requests is dropped in the configuration C3. The maximum waiting time for the requests is observed with configuration C3.

Agent 4					
	Avg Res Time	Avg Serv Time	Avg Wait Time	Pro Idle	Avg drop
C1	0.008	0.007	0.001	0.181	0.004
C2	0.018	0.016	0.002	0.125	0
C3	0.033	0.025	0.008	0.116	0.011
C4	0.032	0.025	0.007	0.116	0.009
C5	0.014	0.013	0.001	0.179	0.004
C6	0.015	0.013	0.002	0.118	0

Table 6: Simulation results obtained for Agent 4 with different Configuration.

The Table 7 presents the results obtained for Agent 5 for the different configurations considered. Agent 5 has taken the maximum time 0.07 to respond under Configuration C4 and the minimum time to respond is 0.016 with configuration C1 and C5. The maximum number of requests is dropped in the configuration C4. The maximum waiting time for the requests is observed with configuration C4.

It is observed that the response times of Agent 3, Agent 4 and Agent 5 in the considered configuration are closer to each other and only a few numbers of negotiation services are dropped. Maximum response time is experienced with Configuration C3 and C4 and observed that the dropping of requests in the Internet is the least for the configuration C3 and C4. Lowest Response time is in C1 for Agents A1, A3, A4, A5 but experienced highest number of dropping in the requests. The behavior of Agent 2 is observed different compared to all the other agents' behavior. The reason can be that the average workload of Agent 2 is less compared to the workload of other agents, because of that Agent 2 could execute all the negotiation

Agent 5					
	Avg Res Time	Avg Serv Time	Avg Wait Time	Pro Idle	Avg drop
C1	0.016	0.013	0.002	0.218	0.005
C2	0.022	0.018	0.004	0.211	0.004
C3	0.025	0.02	0.005	0.107	0.002
C4	0.07	0.045	0.026	0.091	0.03
C5	0.016	0.013	0.003	0.128	0.008
C6	0.025	0.021	0.004	0.126	0

Table 7: Simulation results obtained for Agent 5 with different Configuration.

INTERNET					
	Avg Res Time	Avg Serv Time	Avg Wait Time	Pro Idle	Avg drop
C1	0.0607	0.021	0.586	0.047	0.83
C2	0.088	0.001	0.087	0.051	0.413
C3	0.025	0.021	0.004	0.033	0.068
C4	0.07	0.004	0.066	0.048	0.22
C5	0.163	0.004	0.159	0.05	0.674
C6	0.068	0.004	0.064	0.048	0.198

Table 8: Simulation results obtained for Internet with different Configuration.

requests it received. Also we observed that when the Internet speed is increased the dropping of requests reduced which gives the inference that many negotiation requests are executed by the agents successfully.

3 Summary

In this work, we presented a methodology to model the negotiation between the agents and predicting the performance of the system. We presented methodology to: i) develop a mathematical model for the workload of negotiation scenarios over a time horizon, ii) modeling the execution environment, iii) and iv) analyzing the execution environment for variations in resource configurations. The sensitivity analysis is done by considering modification in the resource configuration one at a time, and it also describes bottleneck resources. The output showed that how the different configurations of resources affect the response time of the agents.

4 References

[1] A.Dorri, S. S. Kanhere and R. Jurdak, "Multi-Agent Systems: A Survey," in IEEE Access, vol. 6, pp.28573-28593,2018. <https://doi.org/10.1109/ACCESS.2018.2831228>

[2] Jing Xie & Chen-Ching Liu (2017) Multi-agent systems and their applications, Journal of

International Council on Electrical Engineering, 7:1,188-97,DOI:10.1080/22348972.2017.1348890 <https://doi.org/10.1080/22348972.2017.1348890>

[3] Wooldridge, M.: An Introduction to Multi-Agent Systems. John Wiley Sons, Inc. New York, NY, USA (2001)

[4] Ebrahim AlHashel, "A Conceptual agent Cooperation Model for Multi-agent Systems' Team Formation Process", Third 2008 International Conference on Convergence and Hybrid Information Technology, pp. 12-20,2008. <https://doi.org/10.1109/ICCIT.2008.367>

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

[6] S. Balsamo, A. D. Marco and P. Inverardi., "Model-Based Performance Prediction in Software Development: A Survey." IEEE Transactions on Software Engineering, May 2004, Vols. Vol. 30, No.5. <https://doi.org/10.1109/TSE.2004.9>

[7] 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. pp. 58-70. <https://doi.org/10.1145/350391.350406>

[8] Ajitha S, Suresh Kumar T.V, Rajanikanth K. A Quantitative Framework for early prediction of Cooperation in Multi-Agent System. ICTACT Journal on Soft Computing 2013; 587-595, DOI: 10.21917/ijsc.2013.0085. <https://doi.org/10.21917/ijsc.2013.0085>

[9] S. Ajitha, Dr.T.V.Suresh Kumar, Dr.K.Rajanikanth, "Artificial Neural Network Approach for predicting performance of MAS using SPE approach". International Journal of Software Engineering, Volume6, No.2 ,pages 3-20, July 2013.

[10] S. Ajitha, Dr.T.V.Suresh Kumar, D.E.Geetha, Dr.K.Rajanikanth "Modeling Co-operative Index of Multi-Agent Systems using Execution Graph". Proceedings of International Conference on Advances computing in Intelligent Systems and Computing Volume 174,2012, pp41-48, Springer,DOI:10.1007/978-81-322-0740-5. <https://doi.org/10.1007/978-81-322-0740-5>

[11] S. Ajitha, Dr.T.V.Suresh Kumar, D.E.Geetha, Dr.K.Rajanikanth "Early Performance Prediction of Co-operative Multi-Agent Systems" procedia Engineering,38(2012)3037-3048,DOI:10.1016/j.proeng.2012.06.354. <https://doi.org/10.1016/j.proeng.2012.06.354>

[12] Ye Chen, Yun Peng, Tim Finin, Yannis Labrou, Bill Chu, Jian Yao, Rongming Sun, BobWillhelm, Scott Cost, A negotiation-based Multi-agent System for Supply Chain Management ,In Proceedings of Agents 99 Workshop on Agent Based Decision-Support for Managing the Internet-Enabled Supply-Chain.

[13] T. Wong, C. Leung, K. Mak, and R. Fung, "An agent-based negotiation approach to integrate process planning and scheduling," International

Journal of Production Research, vol. 44, no. 7, pp. 1331-1351,2006.

<https://doi.org/10.1080/00207540500409723>

- [14] W L Yeung, Performance of Time-Bound Negotiation in Agent-Based Manufacturing Control, Proceedings of the World Congress on Engineering 2012 Vol III WCE 2012, July 4 - 6, 2012, London, U.K.
- [15] D.E Geetha, T.V. Suresh Kumar, Performance Modeling and evaluation of Distributed Systems, Ph.D thesis, Visvesvariah Technological University, Karnataka, 2012