International Journal of Computer Networks and Communications Security

VOL. 3, NO. 8, AUGUST 2015, 337–342
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Fhorganization; New Organization Model for Multi-Agent Systems

Amin Rahmanzadeh[1] and Eslam Nazemi[2]

[1, 2] Self-* Laboratory, Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran Iran

E-mail: [1]amin.rahmanzade@gmail.com, [2]nazemi@sbu.ac.ir

ABSTRACT

Nowadays, multi-agent systems are frequently used to control big components of IT systems. Since this systems are big and use and produce big data, the control of multi-agents systems by people is not beneficial anymore. So, self-organization, in which agents can organize themselves, is proposed as a resolution. In a self-organizing multi-agent system, the desired behavior of system is emerged based on local behavior of agents. But, emergence is not always as we desired and can result in a manner that we do not want the system to have it. On the other hand, self-adaptation is a centralized and top-down process, in which a central control loop can monitor, analyze, plan and execute decisions on a controlled element. Therefore, a combination of self-organization and self-adaptation can help the system stay self-organized while the bad results of emergence are controlled by self-adaptation features. Recently, combining self-organization and self-adaptation features is becoming a trend. In this paper, we tried to propose an organization model for self-organizing multi-agent system based on self-adaptation features to control emergence.

Keywords: *Self-Organization, Self-Adaptation, Multi-Agent Systems, Organization Model, Organization in Multi-Agent Systems.*

1 INTRODUCTION

In a multi-agent and distributed system, agents' cooperation has a major effect on non-functional requirements of it. Using organizations is a way of coordination in these systems. It means each agent can be assigned to a role [6]. Generally, organization in multi-agent systems is a set of agents contacting based on a set of roles to achieve to a general goal. According to [7], organization in multi-agent system is a way of controlling these systems and according [2], using organization in helps agents do big and complex jobs while so simple themselves. Therefore, self-organizing multi-agent systems are considered an effective solution.

In many of researches, self-organization and emergence are assumed equal. But this idea is wrong. These two concepts refer to two different aspects of a distributed system and can exist next to each other [13]. Based on [14], Self-organization refers to a dynamic and spontaneous re-organization. Components of a self-organizing System, for example agents in multi-agent systems, behave autonomously. The result of this behavior is a general behavior of the system. In another words, the flow of control is bottom-up. On the other hand, emergence is defined as a structure or framework, a behavior, or a function of a system. An emergence has at least two levels: macro and micro levels that have mutual dependencies. Macro level limits micro level and micro level causes the macro level. Based on these two definitions, we can say emergence is the way that self-organization achieves to its goals through. The main motivation of this paper, is that the emergence can get in a situation that is out of control [13, 14]. It means the general behavior of the system can be a behavior that agents, in the micro level, did not mean to make.

Based on [8], a self-adaptive system should be able to do four steps on its controlled element. It should be able to monitor the element, after gathering required information, analyze them, plan on what to do next based on knowledge base (to re-configure, heal, protect, or optimize the element), and execute the made decisions.

In our proposal model, we have tried to merge self-adaptation and self-organization features and omit self-organization problems, including emergence. As we said before, a self-organizing system uses emergence to achieve its goals. Agents have their own local goals which are in line with the main goal of the whole system. In another words, behavior of agents in lower level, causes main behavior in upper level and the so-called emergence phenomena happens. Since, the main motivation of this paper, is that the emergence can get out of control and the general behavior of the system can be an undesired behavior that agents, in the lower level, did not mean to make, studying self-adaptation features, we can concluded that the top-down control flow of self-adaptation can be used for controlling emergence phenomena. This is a new trend and many works are done in this area. In fact, we tried to propose a model that provides the self-adaptation features by having the system organize itself in a manner that does not damage the bottom-up process of self-organization.

Kasinger and et al [12] proposed an organization model for self-organizing multi-agent systems based on self-adaptation features. We will prove that our model improves the response time and consequently, the organization process time. Therefore, the rest of paper is organized as the following: in section 2 we will introduce and elaborate our proposed organization model. Section 3 is about the model we will simulate and compare to our model. In section 4, we will talk about simulation and evaluation models and response time in both of them. Conclusion and references are in section 5 and 6.

2 PROPOSED MODEL

We introduced our model previously in [19]. As we said before, the goal is to propose an organization model in which a self-organizing multi-agent system can have self-adaptation features as well. Therefore, an agent should have control role and implement the control loop. The controlled element would be the whole system. This agent should be able to gather information from other agents, analyze them, and send back some non-controlling knowledge to them. Agents can use this knowledge to achieve to self-organization goal, without involving to a chaos. In other words, the goal is emerged. So, the first requirement is to have a tree like structure in which one agent controls other ones.

In this kind of one controller agent, that agent may become a bottleneck. Distribution of controlling role to multiple levels can be a solution. In fact, agents in different levels can take part in controlling responsibility. In this case, we have a complete hierarchical structure.

In a complete hierarchical structure, there is no way for grouping agents based on application and scenario of the system. Grouping makes the possibility to have all kinds of scenarios and applications. In this situation, agents of one group can participate in the hierarchy structure by means of a delegate. This type of grouping is called federation. Therefore, our model is a combination of Federation and Hierarchy. We call this model FHOrganization. Different paradigms of organization, including hierarchy and federation are introduced in [2].

So far, we mentioned general features of FHOrganization model. FHOrganization can make adding of self-adaptive control loop to self-organizing multi-agent systems possible. Table 1. shows these features and equivalent features in federation and hierarchy paradigms.

Components of FHOrganization are showed in Fig. 1 As you can see, an FHOrganization is consist of one or more Agent(s), one or more Group(s), and one and only one Controller Agent. Each Agent is connected to one and only one Group and each Group is consist of one or more Agent(s). Each Group has one and only one Delegate Agent and each Delegate Agent is connected to one and only one Middle Agent. Each Middle Agent can be connected to zero or more Delegate Agent(s). Middle Agents may not be connected to Controller Agent but Controller Agent is connected to one or more Middle Agent(s).

Table 1: main features of FHOrganization

Required features	Equivalent feature
A tree like structure with one agent on top	Hierarchical organization: Agents are organized in a tree like structure in which agents in higher level have more comprehensive view on agents in lower level [2].
Possibility of more than one level to distribute controlling role	Hierarchical organization: Data produced in lower level goes up to make a wider view of all agents. Meanwhile, commands, made based on received

	data, go down (showing existence of different levels) [2].
Existence of agents as groups delegate	Federation organization: A group of agents give part of their autonomy to a delegate. This delegate is responsible to contact outside the group [2].

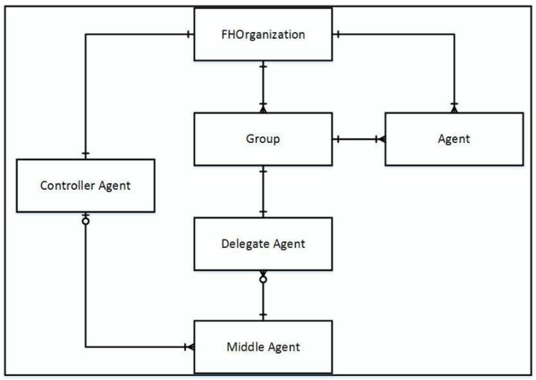

Fig. 1. Components of FHOrganization Model

In this part, for a better understanding of FHOrganization, we bring an example of how this model shapes the organization of a multi-agent system. As you can see in Fig. 2, this scenario is consist of 13 agents distributed in an environment. This Figure shows the practical distribution of agents.

In this example, agents are grouped in four A, B, C and D groups due to their distance (metrics can defer based on scenario). Each group has a delegate. Delegate of A is A3, delegate of B is B4, delegate of C is C3, and delegate of D is D3.

There are two type of relations among agents. First type showed as dotted arrows is established among agents from one group. Second type showed as complete arrows is established among agents from two different groups. But you may ask: why there is why C1 and C2 have second type relation? The answer is in Fig. 3.

Fig. 3 shows the hierarchical structure of FHOrganization. This figure is a hypothetical illustration of the scenario. In this scenario, organization is established in three levels. Agents are grouped in lower and upper level (hierarchy of federation). Therefore, there is a controlling structure with distributed control in different levels. X, Y and Z dimensions are placed in the figures to have a better understanding.

3 ONE SUPERVISOR ORGANIZATION

As we said before, kasinger and et al proposed a model for self-organizing multi-agent systems in which there is a supervisor agent in the top of other agents [12]. In this section, we will briefly describe the proposed model. We have chosen this model to comparison with our model. Because this is the closest model to ours that we found. We call this model One Supervisor Organization or OSOrganization.

In this model, one of the agents is chosen as the supervisor agent. This agent uses the control loop in it to detect repeated problems in the system and use exception rules to solve them. This structure is proposed because of the reasons we mentioned before. In the next section, we will propose our implementation of both models to compare their response time. Fig. 4 shows the structure of the supervisor agent in OSOrganization.

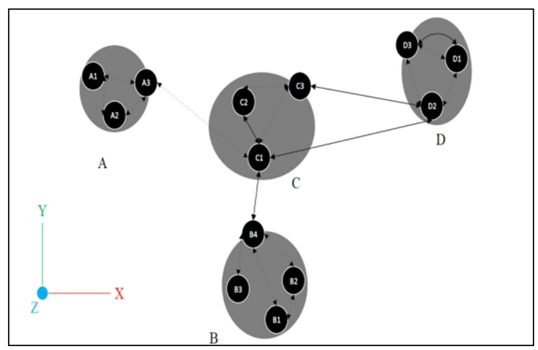

Fig. 2. Practical Schema of Hypothetical Scenario

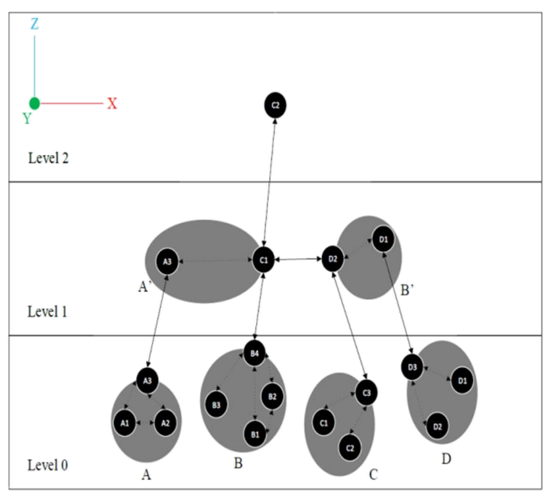

Fig. 3. Hypothetical Scenario with FHOrganization

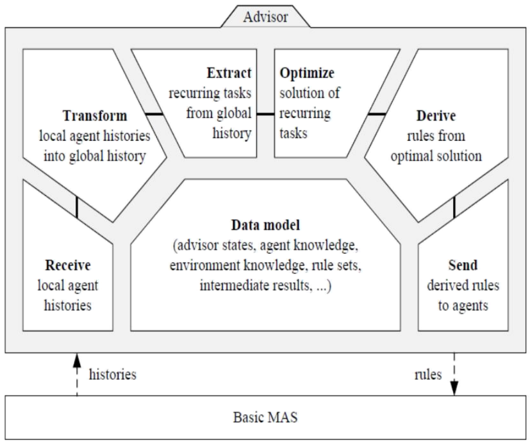

Fig. 4. Structure of Advisor Agent in Model Proposed in [12]

4 SIMULATION AND EVALUATION

In this section, implementation and evaluation of FHOrganization will be presented. We will define a scenario and compare both FHOrganization and OSOrganization response time in that scenario. Implementation is done using Java Agent Development framework (JADE).

In our hypothetical scenario, 50 agents are distributed in a two dimension environment. One agent is going to send a massage to supervisor agent and then the supervisor will send it to another agent. First of all the structure should be established. Pseudo code for both FHOrganization and OSOrganization models are presented in Table 2.

Table 2. Pseudo Code of Establishing FHOrganization and OSOrganization

1. Random(X,Y); 2. Detect(MiddlePoint); 3. Detect(Group); 4. Determine(SupervisorAgent); 5. Determine(DelegateAgents); 6.Introduce(SupervisorAgent, DelegateAgents); 7.Introduce(DelegateAgents, RegularAgent);	FHOrganization
1. Random(X,Y); 2. Detect(MiddlePoint); 3. Determine(SupervisorAgent);	OSOrganization

The scenario is performed for 50 times in both organization models. Fig. 5 shows the result of response time in FHOrganization. On the other hand, fig. 6 shows response time in OSOrganization. Finally, fig. 7 and 8 show comparison between response time and average response time of both models. As you can see, response time of FHOrganization is better than that of OSOrganization in almost every single try. As you can see, average response time in FHOrganization is 2.94 (ms) and that in OSOrganization is 20.62 (ms).

5 REFERENCES

In this paper, we proposed a model that adds self-adaptive features to self-organizing multi-agent systems. This model keeps distributed and bottom-up characteristic of self-organizing systems while adding centralized and top-down self-adaptation to it. This model provides a situation to control emergence by using self-adaptive control loop.

We also simulated and implemented this model using JADE and compared it to another model which is proposed with the same purpose. The results show that the response time of our model is better than that of the other model.

One of our future works will be developing a framework based on JADE that has the desired feature of having self-adaptation and self-organization as a default. Also, having specific self-healing, self-optimizing, self-protecting and self-configuring features in t our model can another future work.

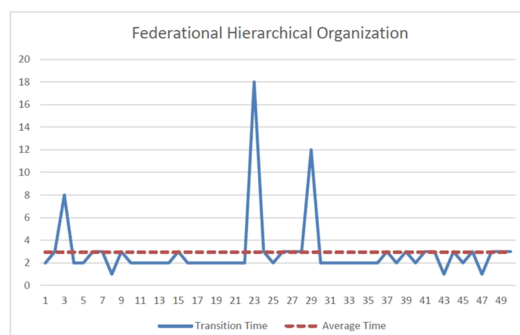

Fig. 5. Response Time in 50 Try in FHORganization

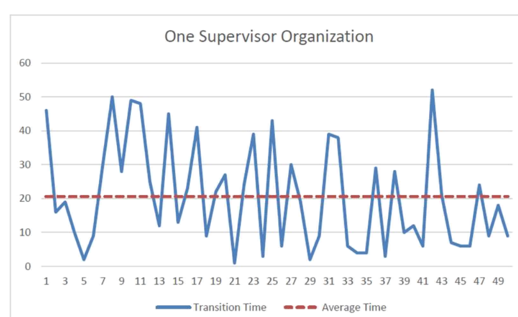

Fig. 6. Response Time in 50 Try in OSOrganization

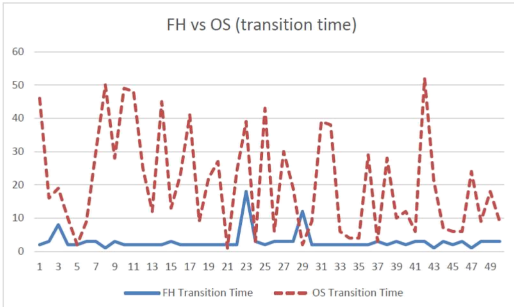

Fig. 7. Comparison between Response Times

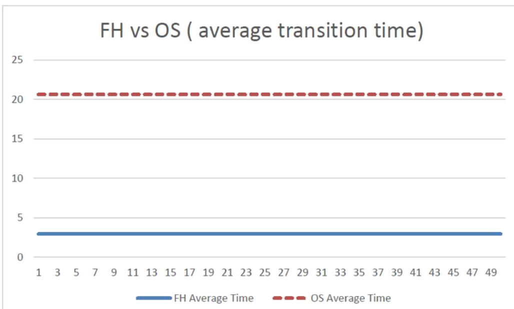

Fig. 8. Average Response Times

6 REFERENCES

[1] Ferber, Jacques, and Olivier Gutknecht. "A meta-model for the analysis and design of organizations in multi-agent systems." In Multi Agent Systems, 1998. Proceedings. International Conference on, pp. 128-135. IEEE, 1998.

[2] Horling, Bryan, and Victor Lesser. "A survey of multi-agent organizational paradigms." The Knowledge Engineering Review 19, no. 04 (2004): 281-316.

[3] Rafi, Kamaleddin Yaghoobi, Ali Farahani, and Eslam Nazemi. "An organizational model for autonomic intelligent distributed systems." Global Journal on Technology 1 (2012).

[4] Ferber, Jacques, Olivier Gutknecht, and Fabien Michel. "From agents to organizations: An organizational view of multi-agent systems." In Agent-Oriented Software Engineering IV, pp. 214-230. Springer Berlin Heidelberg, 2004.

[5] Kota, Ramachandra, Nicholas Gibbins, and Nicholas R. Jennings. "Self-organising agent organisations." In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, pp. 797-804. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[6] Weyns, Danny, Robrecht Haesevoets, Alexander Helleboogh, Tom Holvoet, and Wouter Joosen. "The MACODO middleware for context-driven dynamic agent organizations." ACM Transactions on Autonomous and Adaptive Systems (TAAS) 5, no. 1 (2010): 3.

[7] Dignum, Virginia, Frank Dignum, and Liz Sonenberg. "Towards dynamic reorganization of agent societies." In Proceedings of Workshop on Coordination in Emergent Agent Societies at ECAI, pp. 22-27. 2004.

[8] An architectural blueprint for autonomic computing, IBM Group, 2003,

[9] Truszkowski, Walt, Mike Hinchey, James Rash, and Christopher Rouff. "NASA's swarm missions: The challenge of building autonomous software." IT professional 6, no. 5 (2004): 47-52.

[10] Mathieu, Philippe, Jean-Christophe Routier, and Yann Secq. "Dynamic organization of multi-agent systems." In Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, pp. 451-452. ACM, 2002.

[11] Kolp, Manuel, Paolo Giorgini, and John Mylopoulos. "Multi-agent architectures as organizational structures." Autonomous Agents and Multi-Agent Systems13, no. 1 (2006): 3-25.

[12] Kasinger, Holger, Bernhard Bauer, Jörg Denzinger, and Tom Holvoet. "Adapting environment-mediated self-organizing emergent systems by exception rules." In Proceedings of the second international workshop on Self-organizing architectures, pp. 35-42. ACM, 2010.

[13] De Wolf, Tom, and Tom Holvoet. "Emergence versus self-organisation: Different concepts but promising when combined." In Engineering self-organising systems, pp. 1-15. Springer Berlin Heidelberg, 2005.

[14] G. D. M. Serugendo, M. P. Gleizes, A. Karageorgos, “Self-Organization and Emergence in MAS: An Overview”, Informatica, pp. 41-52, 2006, Slovenia.

[15] Truyen, Eddy, and Wouter Joosen. "Towards an aspect-oriented architecture for self-adaptive frameworks." In Proceedings of the 2008 AOSD workshop on Aspects, components, and patterns for infrastructure software, p. 9. ACM, 2008.

[16] Capera, Davy, et al. "Emergence of organisations, emergence of functions."AISB’03 symposium on Adaptive Agents and Multi-Agent Systems. 2003.

[17] Bellifemine, Fabio Luigi, Giovanni Caire, and Dominic Greenwood. Developing multi-agent

systems with JADE. Vol. 7. John Wiley & Sons, 2007.
[18] Steghofer, J-P., et al. "Improving the Efficiency of Self-Organizing Emergent Systems by an Advisor." Engineering of Autonomic and Autonomous Systems (EASe), 2010 Seventh IEEE International Conference and Workshops on. IEEE, 2010.
[19] Amin Rahmanzadeh, Ali Farahani, Eslam Nazemi, A New Organization Model for Self-Organizing Multi Agent Systems Based on Self-Adaptation Features, International Journal of Intelligent Information Systems. Special Issue:Research and Practices in Information Systems and Technologies in Developing Countries. Vol. 3, No. 6-1, 2014, pp. 15-22. doi: 10.11648/j.ijiis.s.2014030601.13M.
[20] Risoldi, J. L. Fernandez-Marquez, G. Di Marzo Serugendo, "Resilience Framework for self-organising systems", Adaptive, Dynamic, and Resilient Systems, G. Cabri and N. Suri (Eds), Taylor and Francis, 2013.
[21] G. Di Marzo Serugendo, J. L. Fernandez-Marquez, "Self-Organising Services", Int. Conf. on Self-Adaptive and Self-Organizing System, Philadelphia, USA, 2013.
[22] G. Stevenson, D. Pianini, S. Montagna, M. Viroli, J.Ye, S. Dobson, "Combining self-organisation, context-awareness and semantic reasoning: the case of resource discovery in opportunistic networks", Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra (POR) March, 2013.
[23] Klusch, Matthias, and Jörg Müller Huaglory Tianfield. "Multiagent System Technologies."
[24] Kota, Ramachandra, Nicholas Gibbins, and Nicholas R. Jennings. "Decentralized approaches for self-adaptation in agent organizations." ACM Transactions on Autonomous and Adaptive Systems (TAAS) 7, no. 1 (2012): 1.
[25] Gorodetskii, V. I. "Self-organization and multiagent systems: I. Models of multiagent self-organization." Journal of Computer and Systems Sciences International 51, no. 2 (2012): 256-281.

AUTHOR PROFILES:

Amin Rahmanzadeh received the B.Sc. degree in Information Technology Engineering from Tabriz University, in 2012 and the M.Sc. degree in Information Technology Engineering – Enterprise Architecture from Shahid Beheshti University, in 2015. Currently, he is a Research Assistant at Self-* Laboratory, Shahid Beheshti University. His interests are in Autonomous Computing, Self-Adaptation and Self-Organization and Enterprise Architecture.

Dr. Eslam Nazemi received the B.Sc. degree in Computer and Research in Operation from former Planning and Computer Application School in 1977 and two M.Sc. degrees in Systems Engineering (Minor: Research in Operation) and Systems Engineering (Minor: Economy) from Isfahan University of Technology and Research and Development Institute in 1989 and 1997. He received the Ph.D. degree in Industrial Engineering (Minor: Information Technology) from the Islamic Azad University (Science and Research Branch). Currently, he is a professor at Shahid Beheshti University. His research interests include Autonomous Computing and Self-* concepts.