IJICTR
International Journal of Information & Communication Technology Research

Volume 11- Number 4 – Autumn 2019 (29 -39)

A Model for Selecting the Best Organizational Paradigm at Run Time for Self-Organizing Multi-Agent Systems; an IoT Case Study

Amin Rahmanzadeh
Faculty of Computer Science and Engineering
Shahid Beheshti University G.C (SBU)
Tehran, Iran
a_rahmanzadeh@sbu.ac.ir

Mastooreh Rahmani
Faculty of Computer Science and Engineering
Shahid Beheshti University G.C (SBU)
Tehran, Iran
mas.rahmani@mail.sbu.ac.ir

Eslam Nazemi*
Faculty of Computer Science and Engineering
Shahid Beheshti University G.C (SBU)
Tehran, Iran
nazemi@sbu.ac.ir

Received: 2 April 2019 - Accepted: 16 August 2019

***Abstract*—Nowadays, using Multi-Agent Systems (MASs) as a way of controlling complex and large-scale systems is becoming very popular. Also, since the scale of the systems is growing up and their environmental changes are becoming very fast and complicated, the experts are trying to enable these systems to control themselves, instead of having them controlled by humans. One way to devolve this responsibility to the systems is to use self-organization in MASs. To have a self-organizing MAS, agents should be able to shape up an organization. There are different organizational paradigms to be used in self-organizing MASs. Normally, the selection of organizational paradigm is done on design time by the designer of the system. But, in a rapidly changing and complicated environment, the selected paradigm might no longer be suitable for the system. In such a situation, there should be another way for the system to select a new suitable organizational paradigm at runtime. There are some works that provide a comparison among different organizational paradigms based on the performance of the MAS that uses that paradigm. But the comparison is done after the system is designed to have the paradigm. These works do not provide a mechanism for the system to select its paradigm at runtime. In this paper we propose an organization model for self-organizing MASs that provides these systems with the ability to change their organizational paradigm at run time. This organization model considers an amount of utility as the criterion based on which the currently used paradigm can be changed by the system itself. We simulate and evaluate our model in an IoT scenario. The scenario includes a Smart Home and its application of Comfort of Residences. The results show that changing the organizational paradigm and not sticking to the selected one on design time, gives us a 28% improvement on the utility.**

Keywords*-Multi-Agent Systems; Self-Adaptive Systems; Self-Organizing Systems; Organizational Paradigm; Internet of Things; Smart Home

[*] Corresponding Author

I. Introduction

Multi-Agent Systems or MASs are a very popular solution for realizing a vast variety of applications. Dorri and et al categorize these applications in Computer Networks, Robotics, Modeling, and City and Built-in Environments [1]. Smart Buildings are a sub-category of the "City and Built-in Environments" category[2, 3]. As mentioned in [4], the MASs provide a powerful solution for control related challenges in complex systems. Being economically affordable and geographically distributable, MASs are one of the best solutions to control large-scale systems [5]. Therefore, we can say that MASs cannot be neglected in the field of controlling large-scale and complex systems.

There are different approaches to design a MAS [6-9]. But one of the general approaches that contains organization concepts for MASs, is presented in [7]. In this paper we work on self-organizing MASs and whenever MAS is used, we mean a self-organizing one. Based on this approach, the steps to design a MAS are as the following:

1. Global analysis: in this step, a perception of the problem is obtained and the functional requirements are specified.

2. Organizational design: in this step, the organizational paradigm of the MAS is determined and also agent types and their roles are defined.

3. Designing inner activities of the agents: this step includes the designing of inner functionalities and attributes of the agents.

4. Designing agent interactions: in this step, the connections and interactions of the agents are defined. This is done through designing the communications and the communication protocols and mechanisms.

5. Architecture design: in this step architecture of the agents and the whole system is designed.

In this approach, step 2 is where the designer or developer of the system determines the organizational paradigm of the MAS. The paradigms in this paper refer to organizational paradigms presented in [10].

Based on [11], complex systems are very unpredictable. As we know, every system operates in an environment. When a MAS is designed to control a complex system, that system will be the environment for that MAS. Therefore, the changes in the environment will affect the utility that we expect from the system. In a self-organizing MAS, it is the organizational paradigm that determines the formation and capabilities of the MAS. Therefore, changes in the environment may reach a level that the determined organizational paradigm would be no longer suitable for the MAS. Putting these facts together, brings us with the idea of changing the organizational paradigm at run time. Because, keeping on with one paradigm that is selected by a designer, may not work towards the expected utility of the system.

Based on [12], in a self-organizing MAS, the interactions and collaborations of agents are defined based on roles and organizations. Therefore, in this paper we are working on an organization model for MASs that provides agents with the ability to change their organizational paradigm, while doing their tasks based on the application they are designed for. Many researchers have proposed different organizational paradigms and the process to form that organization, for different situations [13-16]. But enabling the system to select a new paradigm and change its current paradigm at runtime is not covered by researchers before.

To sum up, our proposed organization model provides the MASs with the ability to change their organizational paradigm at run time. In this organization model, the agents will have additional roles alongside their application related roles that enables them to detect the need for changing the organizational paradigm and select a new proper one. They will be able to form a new organization based on the selected new paradigm and reorganize themselves.

The rest of the paper is organized as the following: the next section includes analyzing related works. Next, the basic concepts are discussed. Our proposed organization model is presented afterwards. Simulation and Evaluation comes next and we conclude the paper at the end.

II. Related Works

There are plenty of works that provide an assessment on the suitability of a specific paradigm for a specific application like the one in [17]. But they do not provide a general mechanism for every application. The basic paradigms that we refer to, are presented in [10]. In this paper, authors provide different paradigms, their characteristics and the mechanisms for their formation. Related environmental situations to each paradigm is also presented. Based on this paper, the different organizational paradigms can be modified or combined together to present new paradigms. In our assessment and evaluation of our presented organization model, we use two of the paradigms in this paper (Hierarchy and Federation) and one that is a combination of those two paradigms (Federation-Hierarchy) and is introduced in [16]. But, the proposed model is not limited to a specific paradigm.

The first attempt for enabling agents to gain a formation and keep the organization at run time, can be seen in [18]. In this paper, authors provide a framework for MASs that enables agents to reorganize their organization at run time and call it OMACS. This framework provides a basis for reorganization of the agents and detecting important agents for this reorganization. But, the reorganization process includes only getting back to the initially specified organizational paradigm and selecting a new one is not considered. In our proposed organization model, the agents are able to keep their current organization based on their roles and change the organizational paradigm whenever it is necessary.

The MACODO middleware for MASs is presented in [12]. A part of this middleware is an organization model for MASs that enables agents to reorganize and keep their interactions and communications. A MACODO enabled MAS can also organize itself and maintain its organization and robustness. But, they also do not provide a solution for selecting a new organization paradigm at run time.

Corkill and et al in [19] mention organization as an important mechanism for having a better performance in MASs that work in complex environments. In this paper, the authors explore the effectiveness of two of the organizational paradigms. This assessment is so close to the goal of our paper which is increasing the utility of the MAS. But yet, they also do not provide a runtime selection mechanism for organizational paradigms. Also, their approach is not general and depends on the application of the MAS.

The authors in [20] focus on using MASs in real-time strategy games and the impact of utilizing the organizational paradigms in such environments. They measure the performance of the system while using different organizational paradigms for the system. They do the examination on four different paradigms. This work provides a comparison between different performances of the system using different organizational paradigms. But the paradigms are selected by the designers and applied on design time. It means that the system cannot change and select its paradigm like our model does. Furthermore, the assessment is done for a specific application (real-time strategy games) and does not provide a general solution.

The authors in [21] provide an application specific approach for multi-agent systems to solve the Multi-Robot Observation of Multiple Moving Targets Problem [22]. In this approach, target agents can gain multiple structures or organizational paradigms and observer agents are supposed to be able to monitor them. This approach uses predefined situations in which the target agents can gain a new organizational paradigm and does not provide a mechanism for selecting a paradigm based on the environmental changes.

III. Basic Concepts

Before getting to the Proposed Model section, we will discuss the basic concepts that we used in our model. This organization model is presented for self-organizing MASs. Based on [23], organization in MASs means a set of agents interacting and cooperating together based on a number of roles, to satisfy a specified goal. Using organizations enables a group of simple agents to do big and complicated tasks [10].

Based on [8] A self-organizing MAS gains an organizational paradigm at design time and builds its structure based on that paradigm. As we mentioned in Related Works section, a model that provides the ability to change the organizational paradigm at run time is not available. Also, based on [7], using planning, negotiation and decision techniques in MASs are good solutions to enable agents to decide together on selecting a new paradigm for the system. Therefore, we aim to realize our idea of MASs being able to change their paradigm at runtime, by applying these techniques to MASs. Hence, we will need a decision making mechanism for the agents.

Based on [24], there are different techniques for the agents to communicate and interoperate in a MAS, to make a decision based on their common interests. Some of the coordination protocols for agents are Contract-Net, Blackboard Systems, Negotiation, Market Mechanisms and Voting. Among these protocols, Contract-Net is a protocol that is used as a solution for connection problem. The connection problem is about finding the best agent to do a task. This task can be finding the best organizational paradigm. Therefore, if we generalize this concept, we can use it as a way to find the best organizational paradigm for a MAS.

In this protocol, an agent that wants a mission to be done, is called the manager and potential agents to accomplish that mission are brokers. A manager announces some tasks to be done (here, changing the current paradigm) and brokers propose their proposals if they are able to do that job. If their proposals are approved by the manager, the broker will start the task. It should be noted that these roles are not the primary roles of the agents. Agents have their own primary roles in the application they are designed for. These roles are assigned to the agents when it is necessary to change the current paradigm.

In a Contract-Net protocol, a contract means establishing some rules to have a better management of missions in a changing environment. A mission consists of some tasks that need to be done for that mission to be accomplished. Contracts help an agent to do its tasks and are the basis for coordination among those agents. The Contract-Net protocol consists of a loop of announcement, propose and giving awards.

IV. Proposed Model

In our proposed organization model, we try to enable a MAS to evaluate its current organizational paradigm and change it at runtime, whenever it is necessary. This can be realized by changing the connection and communication of the agents in a MAS. The new paradigm is selected and applied, whenever the current paradigm is no longer satisfying the expected utility of the system. The newly selected and applied paradigm is supposed to take the value of the utility of the system back to the amount that is expected. The utility value of a system can be defined based on different applications and criteria. We will discuss more on utility later in this paper.

For the sake of clarity, we will present our organization model using a scenario. Let's consider a scenario in which our model will engage: a MAS is working and its organization is held upright by its agents, communicating and interacting with each other. As in self-organizing MASs, whenever a change happens in the environment, the agents reorganize to keep their organization (changes can be losing an agent or joining of a new one). But, what if the changes in the environment take the system towards a situation that makes current paradigm not suitable enough? There are several situations in which this can happen. For example, two MASs are going to merge to each other, some agents are broken and cannot work anymore and there are not enough agents to take their roles, there is some kind of competition between agents to use resources, there is an increase in computation load and etc. A solution is to stop the system and change agents' implementation to provide them with the ability to form a new suitable organization. But this solution has the cost of stopping the system. Another solution would be enabling the system with the ability of changing its organizational paradigm at runtime.

So far, our model has some requirements to be realized. Model Requirements and Proposed Solutions. shows the requirements of our proposed model and the equivalent solution we propose for those requirements.

TABLE I. MODEL REQUIREMENTS AND PROPOSED SOLUTIONS.

Model Requirements	Equivalent Solutions
Agents need to communicate	Using coordination-cooperation protocols
Agents need to decide together	Using negotiation protocol
The new paradigm should be suitable for the environment	Using agent-based approach of negotiation protocol
Agents should apply their decision	Using Contract-Net protocol

To apply the Contract-Net protocol and enable the agents to decide on changing and selecting a new organizational paradigm, we proposed an organization model for MASs. Fig. 1 illustrates our proposed model. This model shows the main components of a MAS that is able to change its current organizational paradigm on demand. As you can see, there is a database to record the data about the history of operations, a manager agent and at least one broker agent. The broker agent's mission is to apply the paradigm that is selected by the manager. The manager agent knows about goals of the system and also the environmental changes. It can access the database and select the best organizational paradigm based on those goals and changes. In the following, we provide the description of components in this model:

- Multi-Agent System or MAS: this is the whole MAS and has one Manager Agent, one Broker Agent, lives in one to many Environmental Situations, organizes itself based on selected Organizational Paradigm, and is designed to realize an application and therefore, it has some goals to be satisfied.

- Application and Goals: a MAS is designed to realize one or more applications and each application pursues at least one goal.

- Environmental Situation: as we will see, there may be different Environmental Situations for a MAS. The Broker Agent monitors these situations to calculate utility and update the Database based on them.

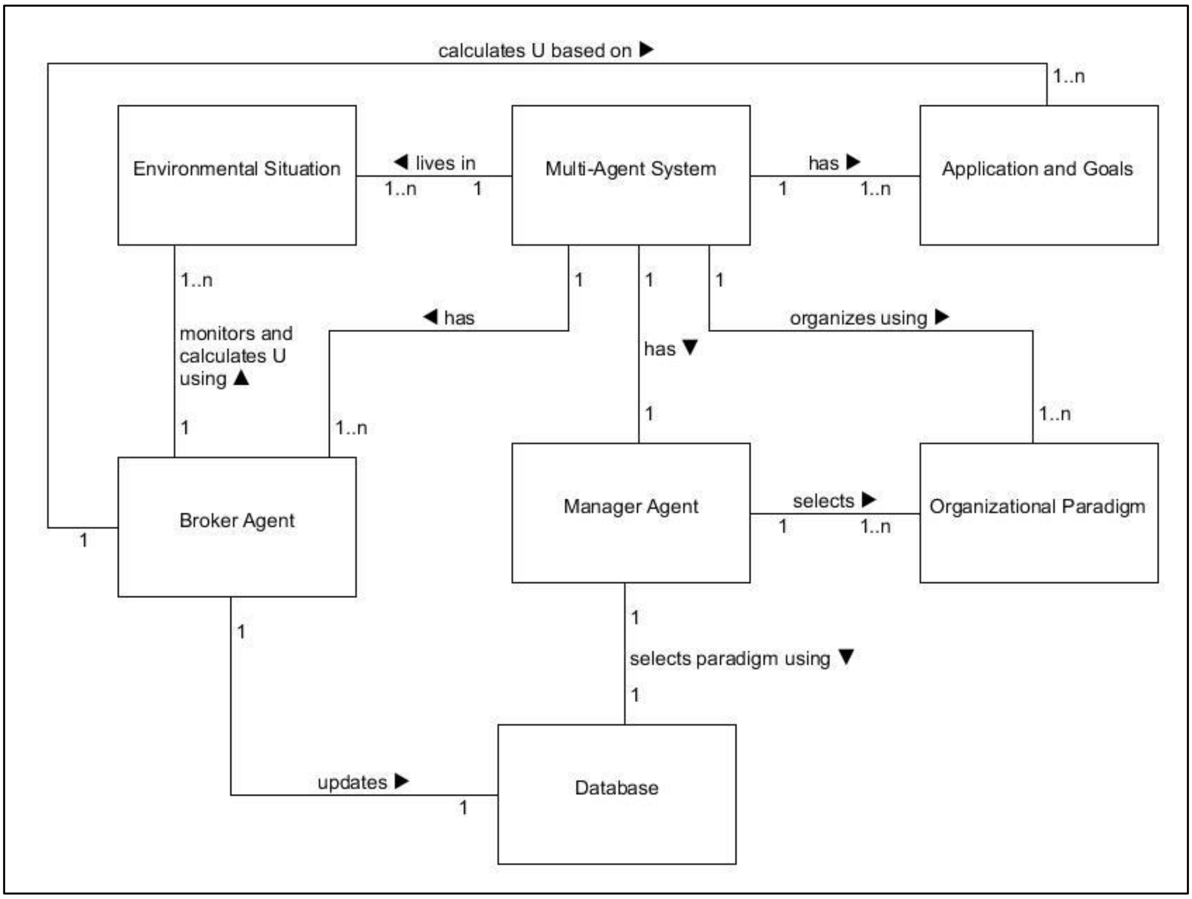

Figure 1. Proposed Organization Model for MASs.

- Manager Agent: the Manager Agent is the one agent that selects Organizational Paradigms for the MAS using information recorded on Database.
- Broker Agent: the Broker Agent is the one that updates the Database. For each Organizational Paradigm, there will be one Broker Agent in the MAS. Assigning both Broker and Manager Agent roles to agents may have different metrics that is not the subject of this paper. We used some metrics that will be described in the Simulation and Evaluation section.
- Organizational Paradigm: The formation and topology of a MAS is determined by an Organizational Paradigm. Organizational Paradigm determines the main roles for agents and the way they can communicate.
- Database: Database keeps the historical information about how the Organizational Paradigms that have been working out for the MAS. This can be preserved by recording the utility of using a paradigm in the Database. The Database can be established by one of the agents or can be remotely available as a service for the agents.

In this model, agents will have access to the database. The database is empty at first and each agent is responsible for writing some information on it. These information are resulted from the activities and interactions of the agents. The manager agent will be using (reading) the database and the broker agents will update it. The database is used in order to help the manager with using and utilizing previous experiences, for selecting a better paradigm, when it is necessary to change the current one. In the beginning of the paradigm change process, when there are some environmental changes that gets the utility of the system to an amount that is lower than the expected threshold, there are no historical information about the utility of the system using other paradigms. But as time goes by and agents update data of the database, new decisions will be made more wisely.

Fig. 2 illustrates a general schema of the tables of the database. As you can see, there is one table for each of the environmental situations and there are two columns in that table. One column referrers to the name (or number) of the paradigm and another one refers to the value of the utility of the system, resulted from using that paradigm. Each of the tables will have the results of using Paradigm Pj in Environmental Situation Ei and the resulted Utility Uij.

For an environmental situation to be recorded in the database, we need to quantify it. As in [25], there are some criteria that can be assumed as environmental situations in a MAS. These criteria are

Environment E_i	
Paradigm P_j	Utility U_{ij}

Figure 2. A general schema of the tables of the database.

suitable for providing a quantitative value for environmental situations:

- Number of agents
- Computation time needed for an agent to do its tasks
- Memory needed for an agent to do its tasks
- States of an agent like idle, waiting, active, etc.
- Connections among agents
- Dependency among agents

In this paper, we focus on the number of agents as the environmental situation.

For more clarification, we provide an example of how the manager uses historical data of the database. For example, after a while that the system is running, tables are filled for environments E1, E2 and E3. For each environment, using paradigm P1 results in utilities U11, U21, U31. The same thing happens for P2 and P3. After a while we will have a database as Fig. 3 that illustrates the tables filled with the values of this example.

Let's suppose as a designer, we design a MAS and put some agents in it. Based on [26], it is a task for the designer to select an organizational paradigm for the system. This organizational paradigm should be selected based on characteristics of the environment in which the system is going to execute. But, based on our organization model, we select one of agents to be a manager and give it the task of management, to change organizational paradigm if needed. The manager has the management role alongside its other roles.

Since we design the system to be a self-organizing MAS, we can comprehend that the agents try their best to have a cooperation to do their task and solve the problem that the system is designed for. Therefore, if the system is in a state that its utility is lower than its expected utility, in terms of a certain amount, it is time for the system to have a new organizational paradigm. In another words, the environment of the system has been changed so much that the current organizational paradigm is no longer suitable enough. This is when a change in organizational paradigm is going to happen.

One of the agents (other than the manager) is selected as the broker. The selection of this agent

E_1	
P_1	U_{11}
P_2	U_{12}
P_3	U_{13}

E_2	
P_1	U_{21}
P_2	U_{22}
P_3	U_{23}

E_3	
P_1	U_{31}
P_2	U_{32}
P_3	U_{33}

Figure 3. Examples of tables in database.

does not have any limitation and it can be any of the agents based on the organizational paradigm. The broker agent also can have the broker role alongside its other roles. This agent is the one that submits the utility of the system to the rows of the database. Each time the broker submits the utility, it actually submits an average of the new utility and previously recorded ones to the database. This way, the history of the system is used in the results.

In the following, we will describe the steps of the process of changing the current paradigm and selecting a new one:

Step one: in the first step, if the manager recognizes that a change in paradigm is required, since there is no history of change of the paradigm, it selects a paradigm randomly. The result may be one of the followings:

1. The resulted utility is worse or cannot satisfy the expected utility. In this state, another paradigm gets selected and system gets back to step one.

2. A better utility is gained and it can satisfy the expected utility. In this state, the new paradigm is confirmed and the system works with it. If new changes happen to the environmental situation and the utility goes down, system enters to step two.

Step two: In this step, the database has some information about utilizing other paradigms for the system. In this situation, the manager compares the resulted utility against the recorded utility in database and the result may be one of the followings:

1. The new environmental situation is recorded before and the manager can easily select the proper paradigm. For example, the value of new environmental situation is Ei and it is recorded in the database. In the table of Ei, the best Utility Uij is related to Pj and therefore, the Pj would be selected.

2. The new environmental situation has not happened before. In this state, the manager should consider the following situations:

1.1. If the new environmental situation has a value between two of the values recorded before, the manager should add the utility of those two environmental values for each paradigm and then select the paradigm with maximum value. Let's assume the new environmental situation has the value Ej and there are tables with values Ei and Ek in the database in such a way that the relation Ei<Ej<Ek is true. In the table Ei, the values Ui1, Ui2 and Ui3 are recorded for P1, P2 and P3. Also in the table Ek, the values Uk1, Uk2 and Uk3 are recorded for P1, P2 and P3. The manager will use the values Ui1+Uk1, Ui2+Uk2 and Ui3+Uk3 and select the paradigm that is related to maximum value. For example, if Ui2+Uk2 is the maximum value (maximum or minimum based on the application), the P2 will be selected and applied to the system. Also a new table will be created for Ej.

1.2. If the new environmental situation has a value more than ever recorded, the last two tables with lesser environment values than new environment value should be considered. Then the manager will add the utility for each paradigm in those tables and select the paradigm with maximum value. Since it can be concluded from the example described in the previous step and for the sake of lack of space, we will not describe this step with an example.

1.3. If the new environmental situation has a value less than ever recorded, the first two tables with greater environment values than the new environment value should be considered. Then the manager will add the utility for each paradigm in those two tables and select the paradigm with maximum value.

The maximum value refers to the value of the utility that should be increased. But the criterion that is used by the system to calculate the utility, may differ based on the application of the MAS. For example, if the utility is performance of the system and the performance means the time duration to get to the desired temperature, as in our scenario that will be explained in the next section, then the time value should be minimum. Therefore, the designer of the system should provide a formula for the utility, based on the criterion that the system is using. Fig. 4 illustrates the flowchart of the process of changing the current paradigm.

V. SIMULATION AND EVALUATION

In this section, we present the evaluation of our proposed organization model by simulating a Smart Home scenario. The scenario is about controlling the temperature of the house. We used Java Agent DEvelopment or JADE framework [27] to simulate the Smart Home scenario. JADE is one of the most used tools in simulation of MASs.

A Smart Home can control many aspects of lives of its residents. One of the applications of the Smart Homes is the comfort of its residents. In this paper, we work on controlling the temperature of the house. Authors in [28] provide a thorough description of this scenario. In this scenario, we consider the number of agents to be the value of the environmental situation or E, paradigms would be P and the time duration to get to desired temperature would be used to calculate the utility or U. As we mentioned before, in this scenario, our goal is to maximize this utility. Therefore, we should provide a formula for the utility based on the time duration. Formula 1 shows this relation between utility and time duration.

Formula 1

$$U = \frac{1}{T} * 100$$

In this formula U is the value of the utility and T is the time duration to get to the desired temperature.

The utility has a reverse relation with the time duration. Because as in our scenario, it is good to decrease the time duration to get to the desired temperature. Then the value is multiplied by 1000 just to get a better visualization in charts.

In this scenario we assume that the MAS can have three different organizational paradigms: Hierarchy, Federation [10], and Hierarchy-Federation [16] Paradigms. These paradigms are defined for the agent as an initial knowledge. To provide this knowledge, we provided each agent with three different behaviors. Each behavior is used to construct the connections of the agents, based on one of the paradigms. Therefore, the agents already know how to construct an organization with any of the three paradigms. The behavior concept is the mechanism in JADE to enable agents with abilities. We used this mechanism to make it easy to add knowledge of any new paradigm. The details of our simulation is described in the following.

In a hierarchical organizational paradigm, the agents are connected together through a tree. Any agent in a higher level has a wider view on the situation of the environment. In this paradigm, the flow of information is bottom-up and the flow of orders is up-down. Fig. 5 illustrates this organizational paradigm.

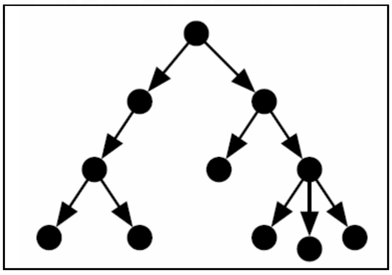

Figure 4. A Hierarchical organizational paradigm [10].

If the system gains the hierarchy organizational paradigm, the chain of commands and information should also be hierarchical. Therefore, the agents placed in leaves of the tree (a hierarchy can be assumed as a tree and the agents in the farthest point from the root, are leaves that have a parent but have no children) give the information of measured temperature to their parents and the parents do the same thing, until the information gets to the root. The root agent has the information about desired temperature. Then the root agent calculates average information using the number of leave agents, compares it with the desired temperature, and if any changes is required, returns the command through the tree to the leave agents. Fig. 6 provides the pseudo code for shaping up a hierarchy organizational paradigm and working towards scenario goals.

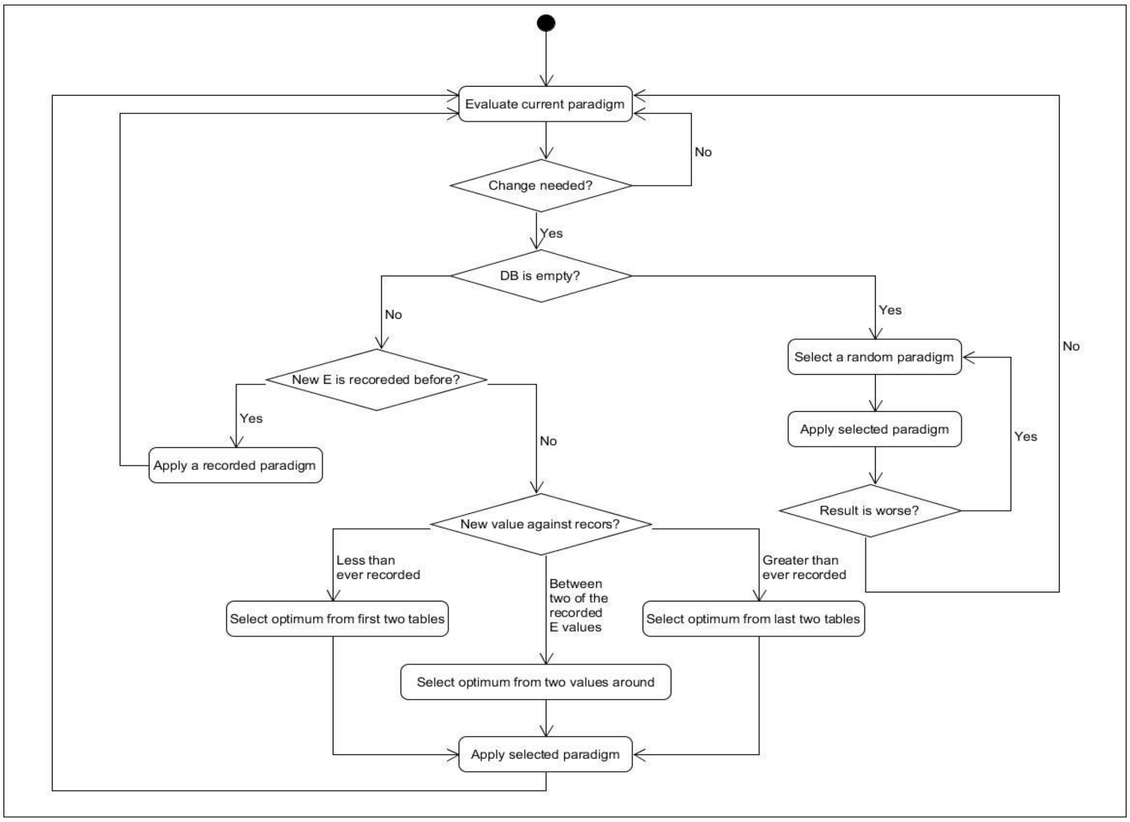

Figure 5. Process of changing the current paradigm.

```
Determine(Root);
Detect(level);
Determine(FatherAgent);
Introduce(RegularAgent, FatherAgent);
RunScenarioUntillBalanced()
```

Figure 6. Pseudo code for shaping up the hierarchy organizational paradigm.

In a federation organizational paradigm, agents are in different groups and give an amount of autonomy to the delegates of those groups. The agents of a group are connected to other groups through the delegate of their group. Fig. 7 illustrates a federation organizational paradigm.

If the system gains federation organizational paradigm, the agents of each group give measured temperature to their delegate and each delegate calculates the average temperature of its group. Then the delegate agent compares the average temperature against the desired temperature and if changes are required, the delegate indicates them to the agents of its group. Fig. 8 provides the pseudo code for shaping up a federation organizational paradigm and working towards scenario goals.

In a federation-hierarchy organizational paradigm, the agents are grouped like the federation organizational paradigm, but the groups also are placed in different levels. This helps the MAS to have benefits of using up-down flow of orders alongside the local decision making of the agents in their groups. Fig. 9 illustrates a federation-hierarchy organizational paradigm.

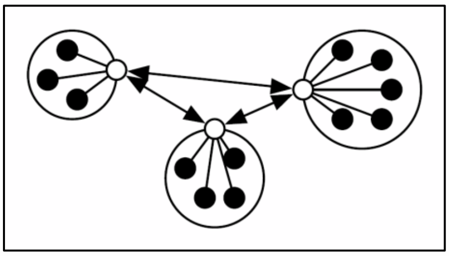

Figure 7. A Federation organizational paradigm [10].

```
Detect(Group);
Determine(DeligateAgents);
Introduce(DeligateAgents, RegularAgents);
Introduce(DeligateAgent, DeligateAgents);
RunScenarioUntilBalanced()
```

Figure 8. Pseudo code for shaping up the federation organizational paradigm.

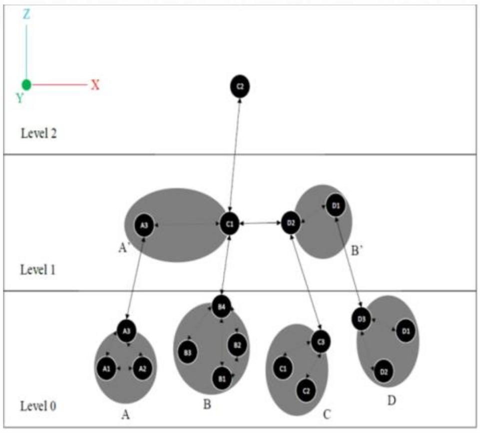

Figure 9. A Federation-Hierarchy organizational paradigm [16].

If the system gains federation-hierarchy organizational paradigm, agents in each group will measure the temperature and give the information to their delegate. The delegate calculates the average and sends the results to the supervisor agent (a role defined for Federation-Hierarchy organizational paradigm). Then the supervisor does the required average calculation and comparison against desired temperature. Fig. 10 provides the pseudo code for shaping up a federation organizational paradigm and working towards scenario goals.

In all three pseudo codes, the final step is RunScenarioUntilBalanced(). Running the scenario until the system is balanced, means for the environment of the system to get to the desired temperature. In the hierarchy paradigm, the broker agent would be the agent in root of tree. In the federation paradigm, the broker agent is (conventionally) one of the delegates. In the federation-hierarchy paradigm, the broker agent is the agent in root. The broker agent is the one that measures time duration of getting to the desired temperature and writes it on the database.

As we mentioned before, there are different criteria that can be supposed as environmental situation in a MAS. In this paper, we chose the "number of agents" criterion to measure. Therefore, when we talk about the value of E, we mean the

```
Detect(Group);
Determine(SupervisorAgent);
Determine(DeligateAgents);
Introduce(SupervisorAgent, DelegateAgents);
Introduce(DeligateAgents, RegularAgents);
RunScenarioUntilBalanced()
```

Figure 10. Pseudo code for shaping up the federation-hierarchy organizational paradigm.

number of agents. We suppose that the system is working in a building with 5 floors and each floor has20 rooms. Based on the desired temperature for each room, the number of agents in each room may be different. The number of sensors and actuators in each room is also 20.

In the beginning of the evaluation, the system starts with one of the paradigms. Agents get organized and when the system starts its application (temperature balancing) the changes are applied to the temperature by agents and meanwhile, the broker agent writes the time duration of getting to the desired temperature on the database. In this phase, we change the number of agents as the environment value. As we mentioned before, if the time duration to get to the desired temperature exceeds the specified threshold, the manager will select a new paradigm randomly. This process keeps going until the suitable paradigm is selected.

For a better understanding of what happens when the paradigm of the system is not changed, we provided a chart for each paradigm to work with different values of E. As you can see in Fig. 11 the horizontal axis shows the number of agents or value of E, the vertical axis shows the utility of the system getting to the desired temperature or U, the values in blue columns show Utilities measured using hierarchy paradigm (U(H), right columns), the values in brown columns show Utilities measured using federation paradigm (U(F), middle columns), and the values in grey columns show Utilities measured using federation-hierarchy paradigm (U(FH), left columns).

To get these results, we tested each paradigm with different number of agents. The numbers are 50, 100, 200, 500, 750, 1000, 1500 and 2000 agents. For each number we executed the tests multiple times to get more accurate results. The average Utility for all paradigms and number of agents, based on Formula 1 is 32. Fig. 11 illustrates that keeping one paradigm in different environmental situations is not always beneficial. As you can see, for example, after changing the value of E from 200 to 500, if the system uses hierarchy paradigm (blue or right columns), the value of utility is decreased suddenly. But if the system could change its paradigm to federation or federation-hierarchy paradigm, it would be much better for the U value.

To test the functionality of our organization model, we considered a building with 50 initial number of agents. The initial organizational paradigm for this MAS is hierarchy paradigm. The desired temperature is 23 Centigrade and the threshold for Utility is upper than 33. We change the number of agents (value of the environment in our scenario) as the input for model. **Error! Reference source not found.** shows our assumptions for the simulation.

We executed the simulation multiple times (1000 runs) and let the broker agents (for each paradigm) measure time duration for the system to get to the desired temperature in different number of agents as

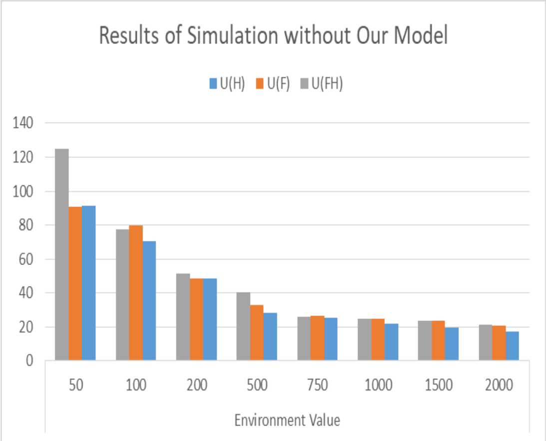

Figure 11. The values of utility when our model is NOT in use.

E and for three different organizational paradigms. The broker agents recorded these information in the database so the manager could use this information in different situations. Fig. 12, Fig. 13, and Fig. 14 show the result of changing paradigms at run time. As you can see, the results, using the proposed model, are always upper than the results without using the proposed model. For example, Fig. 13 compares the value of utility when using proposed organization model against this value when keeping up with the hierarchy organizational paradigm (Brown color or left columns show the value of utility for the proposed model and the blue color or right columns show the value of utility for the hierarchy paradigm)

These figures show that whenever the utility of a paradigm is the maximum one, the manager selects that paradigm and this keeps the utility in the desired threshold. As you can see, the average utility for the system to get to the desired temperature now is 41.

Our model is able to keep the average value of utility over 33 in different number of agents or E. Whenever the utility value goes under 33, the system selects another paradigm and the new paradigm helps system keep the value of time duration under 30.

Also, as we mentioned before, the average utility for the test without our model was 32 and for the test with our model is 41. This shows a 28% improvement in U. Fig. 15 illustrates these results.

TABLE II. ASSUMPTIONS OF THE SIMULATION.

Value	Assumption
Number of agents as E	50, 100, 200, 500, 750, 1000, 1500, 2000
Valid value for Utility to get to the desired temperature as U	>33
Paradigms	Hierarchy, Federation, Federation-Hierarchy
Desired Temperature	23 centigrade
Initial E	50
Initial paradigm	Hierarchy

Number of executions	1000 runs

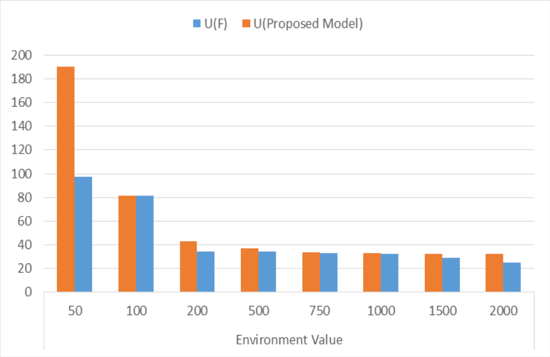

Figure 12. Utility using the proposed model against Utility using only the Federation paradigm.

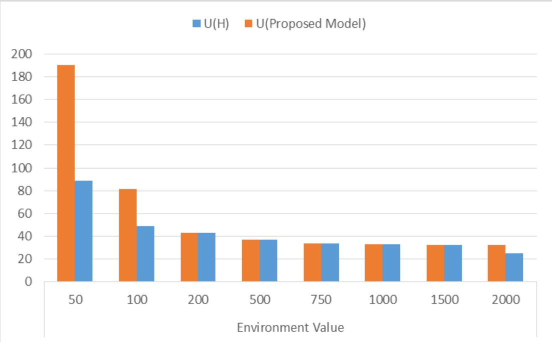

Figure 13. Utility using the proposed model against utility using only the Hierarchy paradigm.

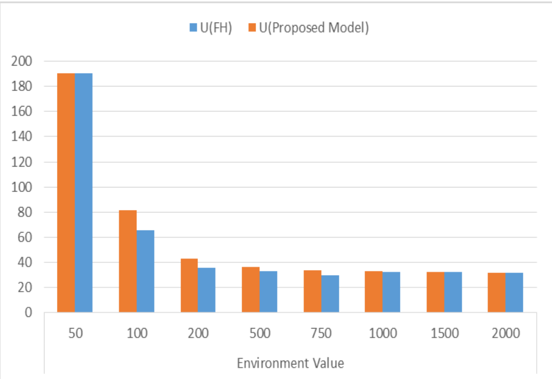

Figure 14. Utility using the proposed model against utility using only the Federation-Hierarchy paradigm.

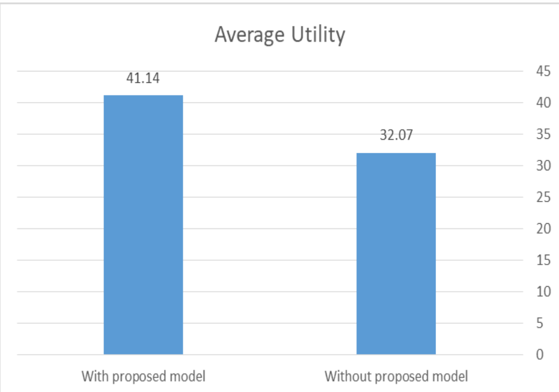

Figure 15. Average utility with and without using proposed model.

VI. CONCLUSION

In this paper we proposed an organization model for Self-Organizing MASs that enables these systems to change their organizational paradigm at run time. This model is inspired from Contract-Net protocol which is a communication and interoperation protocol for agents in MASs. To apply this protocol to our system, we defined two roles for the agents: Manager and Broker agents. Agents can have these roles alongside their other roles that are based on the application that system is designed for.

We designed a Smart Home scenario in which the agents are responsible of controlling the temperature of an environment and used this scenario to evaluate the proposed model. We simulated this scenario using JADE, a popular java framework for agent-oriented development, and gathered the results and compared them with and without using the proposed model.

The results show that with applying our organization model to the MAS, the utility of the system (performance in our scenario) is improved about 28 percent. The model works properly and agents select the best paradigm whenever the changes in the environment are too much that we need to have a new paradigm for the system. The evaluation is done using three organizational paradigms, Hierarchy, Federation, and Federation-Hierarchy paradigms. In the future we are going to test our model using more paradigms and do the evaluation in a real word environment. Also, we are going to enrich our agents' implementation with some initial information on when to select which paradigm using the characteristics of paradigms. This way, the random selection of agents at the beginning with empty database will no longer be necessary.

REFERENCES

[1] Dorri, A., S.S. Kanhere, and R. Jurdak, Multi-agent systems: A survey. IEEE Access, 2018. 6: p. 28573-28593.

[2] Hager, K., J. Rauh, and W. Rid, Agent-based modeling of traffic behavior in growing metropolitan areas. Transportation Research Procedia, 2015. 10: p. 306-315.

[3] van Pruissen, O., A. van der Togt, and E. Werkman, Energy efficiency comparison of a centralized and a multi-agent market based heating system in a field test. Energy Procedia, 2014. 62: p. 170-179.

[4] González-Briones, A., et al., Multi-agent systems applications in energy optimization problems: A state-of-the-art review. Energies, 2018. 11(8): p. 1928.

[5] Stone, P. and M. Veloso, Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 2000. 8(3): p. 345-383.

[6] Qin, J., et al., Recent advances in consensus of multi-agent systems: A brief survey. IEEE Transactions on Industrial Electronics, 2016. 64(6): p. 4972-4983.

[7] Wooldridge, M. and N.R. Jennings, Intelligent agents: Theory and practice. The knowledge engineering review, 1995. 10(2): p. 115-152.

[8] Kinny, D., M. Georgeff, and A. Rao. A methodology and modelling technique for systems of BDI agents. in European Workshop on Modelling Autonomous Agents in a Multi-Agent World. 1996. Springer.

[9] Akbari, O.Z., A survey of agent-oriented software engineering paradigm: Towards its industrial acceptance. International Journal of Computer Engineering Research, 2010. 1(2): p. 14-28.
[10] Horling, B. and V. Lesser, A survey of multi-agent organizational paradigms. The Knowledge engineering review, 2004. 19(4): p. 281-316.
[11] Heylighen, F., Complexity and self-organization. Encyclopedia of library and information sciences, 2008. 3: p. 1215-1224.
[12] Weyns, D., R. Haesevoets, and A. Helleboogh, The MACODO organization model for context-driven dynamic agent organizations. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2010. 5(4): p. 1-29.
[13] Esmaeili, A., et al., The impact of diversity on performance of holonic multi-agent systems. Engineering Applications of Artificial Intelligence, 2016. 55: p. 186-201.
[14] Damba, A. and S. Watanabe. Hierarchical control in a multiagent system. in Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007). 2007. IEEE.
[15] Brooks, C.H. and E.H. Durfee, Congregation formation in multiagent systems. Autonomous Agents and Multi-Agent Systems, 2003. 7(1-2): p. 145-170.
[16] Rahmanzadeh, A. and E. Nazemi, Fhorganization; New Organization Model for Multi-Agent Systems. International Journal of Computer Networks and Communications Security, 2015. 3(8): p. 337-342.
[17] Tchappi, I.H., et al., A brief review of holonic multi-agent models for traffic and transportation systems. Procedia computer science, 2018. 134: p. 137-144.
[18] DeLoach, S.A., OMACS: A framework for adaptive, complex systems, in Handbook of research on multi-agent systems: Semantics and dynamics of organizational models. 2009, IGI Global. p. 76-104.
[19] Corkill, D.D., D. Garant, and V.R. Lesser. Exploring the effectiveness of agent organizations. in International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems. 2015. Springer.
[20] Bernstein, B.A., J. Geurtz, and V.J. Koeman. Evaluating the Effectiveness of Multi-Agent Organisational Paradigms in a Real-Time Strategy Environment: Engineering Multiagent Systems Track. in Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 2019. International Foundation for Autonomous Agents and Multiagent Systems.
[21] Silva, T., et al. Classifying Organizational Structures on Targets in the Cooperative Target Observation. in Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional. 2020. SBC.
[22] Luke, S., et al. Tunably decentralized algorithms for cooperative target observation. in Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. 2005.
[23] Dignum, M., E. Sonenberg, and F. Dignum. Dynamic reorganization of agent societies. in Proceedings of workshop on coordination in emergent agent societies. 2004.
[24] Weiss, G., Multiagent systems: a modern approach to distributed artificial intelligence. 1999: MIT press.
[25] Nagwani, N.K. Performance measurement analysis for multi-agent systems. in 2009 International Conference on Intelligent Agent & Multi-Agent Systems. 2009. IEEE.
[26] Ye, D., M. Zhang, and A.V. Vasilakos, A survey of self-organization mechanisms in multiagent systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016. 47(3): p. 441-461.
[27] Bellifemine, F.L., G. Caire, and D. Greenwood, Developing multi-agent systems with JADE. Vol. 7. 2007: John Wiley & Sons.
[28] Diaconescu, A., et al. Goal-oriented holonics for complex system (self-) integration: Concepts and case studies. in 2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO). 2016. IEEE.

Amin Rahmanzadeh received his B.Sc. degree in Information Technology Engineering from Tabriz University, in 2012 and his M.Sc. degree in Information Technology Engineering – Enterprise Architecture from Shahid Beheshti University, in 2015. Currently, he is a Ph.D. candidate in Software Engineering and a Research Assistant at Self-* Laboratory at Shahid Beheshti University. His interests are in Autonomous Computing, Self-Adaptation and Self-Organization, Multi-Agent Systems, and Internet of Things.

Mastooreh Rahmani received her B.Sc. degree in Software Engineering from Razi University, in 2015 and her M.Sc. degree in Information Technology Engineering – Enterprise Architecture from Shahid Beheshti University, in 2018. Her interests are in Autonomous Computing and Self-Organizing Multi-Agent Systems.

Eslam Nazemi was born in Sarab, Iran, in 1954. He got the B.Sc. degree in Applied Mathematics and Operational Research from School of Planning and Computer Application, Tehran, Iran in 1977, The M.Sc. degree in both System Engineering and Economics in 1987 and 1996, and P.D. in Industrial Engineering and Information Technology in 2005, Iran. He was the faculty Member from 1978 to the present. He has been with the Computer Engineering Faculty at Shahid Beheshti University(SBU), Tehran, Iran. He was the deputy of graduate and education affairs, manager of Informatics Development of Education in SBU and now is the manager of Electronic Education Center in SBU. He is an Associate Professor of Computer Engineering and IT Department. His main fields of research are Self-* Software Engineering, Large Scale Software Development, and Self- Adaptive Software Quality. He has authored and co-authored more than 120 articles in Journals and Conferences and has 10 books on mathematics, project management, software engineering, Software Quality and game theory.