12-2009

REORGANIZATION OF MASSIVE MULTIAGENT SYSTEMS: MOTL/O

Aruntej Seelam
Southern Illinois University Carbondale, aruntej.s@gmail.com

REORGANIZATION OF MASSIVE MULTIAGENT SYSTEMS: MOTL/O

By

Aruntej Seelam

B.E. in Electronics and Instrumentation Engineering
Anna University, 2007

A Thesis
Submitted in Partial Fulfillment of the Requirements for the
Master of Science Degree

Department of Computer Science
in the Graduate School
Southern Illinois University Carbondale
December 2009

THESIS APPROVAL

REORGANIZATION OF MASSIVE MULTIAGENT SYSTEMS: MOTL/O

By

Aruntej Seelam

A Thesis Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Computer Science and

Engineering

Approved by:

Dr. Henry Hexmoor

Dr. Bidyut Gupta

Dr. Wen Chi Hou

Graduate School
Southern Illinois University Carbondale
September 03, 2009

AN ABSTRACT OF THE THESIS OF

ARUNTEJ SEELAM, for the Master of Science degree in Computer Science, presented on September 03, 2009 at Southern Illinois University Carbondale.

TITLE: REORGANIZATION OF MASSIVE MULTIAGENT SYSTEMS: MOTL/O

MAJOR PROFESSOR: Dr. HENRY HEXMOOR

MOTL/O embodies the MOTL paradigm and models organizational adaptation. We report progress on developing computational tools for systematically altering organizational components. This adds a novel dimension to MOTL (Hexmoor, et.al., 2008). This extension is necessary to allow communities of agents or robots to reconfigure their organizational structure in response to changes in the environment. We will explore principled mechanisms for converting a hierarchical organization to an edge type organization. We move toward a solution for organizational adaptation. Beyond current project, many other types of organizational adaptation are possible and require much further research that we anticipate for our future work

AKNOWLEDGEMENTS

I would like to thank Dr. Henry Hexmoor, my advisor, for his advice and support as I developed this thesis. I would also like to thank Dr. Bidyut Gupta and Dr. Wen-Chi Hou for their time and support, as well as the entire Department of Computer Science a Department for their diligent instruction. I would like to express my gratitude to Mr. Brian McLaughlan and Mr. Naren Bhosle for their valuable inputs and support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

MOTL/O embodies the MOTL paradigm and models organizational adaptation. We report progress on developing computational tools for systematically altering organizational components. This adds a novel dimension to MOTL (Hexmoor, et.al., 2008). This extension is necessary to allow communities of agents or robots to reconfigure their organizational structure in response to changes in the environment. Traditional approach of a hierarchical command and control (C2) structure is ineffective (Alberts & Hayes, 2003). Recently, an edge organization has been proposed as a more suitable alternative Command and control structure in the current information age, due to its empowerment of the edge members, better shared awareness among all the members in the organization, interoperability and most importantly, agility and adaptability to dynamic situations (Chang, 2005). We will explore principled mechanisms for converting a hierarchical organization to an edge type organization. Other than structural differences, organizations differ in information flow network and information sharing strategies. We move toward a solution for organizational adaptation. Beyond current project, many other types of organizational adaptation are possible and require much further research that we anticipate for our future work. This task will lay the foundation for automatic organizational adaptation.

This report begins by outlining related work and background in section 2. In section 3 we present an approach to reorganization. Section 4 describes an implementation of a simulated test bed that will help us validate our salient concepts. Results and conclusions are in sections 5 and 6.

CHAPTER 2

BACKGROUND AND RELATED WORK

In highly dynamic operational environments, it is desirable for multi-agent systems (MAS) to be capable of self-directed structural reorganization. The organizational structure of MAS, whether based on a graph, hierarchy, federation, or other form, dictates the communication interactions between agents as well as the distribution of roles and authority throughout the system. When motivated to adapt, agents should do so without human intervention and in a manner that improves the overall performance of the system. Several motivating factors for reorganization will be discussed here as will current proposed methods for performing dynamic reorganization in MAS.

2.1 Current Research Area

Current research in the area of dynamic reorganization in multi-agent systems has yielded many approaches useful in dealing with the problem of adaptation in uncertain and often hostile environments. Given the intelligent and autonomous nature of agents in MAS, individual agents must be capable of locally adapting their interconnection schemes with respect to other agents in the MAS. In addition, agents must be able to accept new roles and to comply with any restrictions or laws associated with these roles. Limitations of communication bandwidth, imposed by the environment or power considerations, or explicitly mandated due to security concerns, implies that agents may not, and probably will not, have a complete picture of the effectiveness and efficiency of the global system.

With these limitations in mind, agents must adapt based on local perception of global performance (M. Gaston and M. desJardins, 2005).

The impetus directing reorganization varies widely. Common adaptation triggers are based on estimates of the overall performance of the MAS, timelines specifying that reorganization should take place at scheduled intervals, or structural requirements. Matson and DeLoach describe other adaptation triggers related to roles and contrast adaptation for efficiency (timeline-based) and for effectiveness (quality-based) (Matson and Deloach, 2004). In this work, the authors illustrate three scenarios resulting in a need for reorganization. The first of these relates to the situation in which the organizational objective demands a role that has not yet been assigned. In this situation, there may or may not be extra agents available to accept the required role. Second, a role that is currently assumed by an agent may be relinquished by that agent, resulting in incomplete role distribution as with the first scenario. Third, an agent may be forced to relinquish a role due to some internal fault or as a result of malicious activity. In this case, the system may not be informed of the need to reassign the lost role. In (Chaimowicz, et. al. 2002), similar triggers are described. These include allocation, reallocation, and exchange. In an allocation scenario, an agent has completed its task and is allocated a new task. In the reallocation scenario, an agent prematurely terminates its current task and is allocated a new one. In the final scenario, exchange, two agents swap tasks. Regardless of the adaptation trigger driving reorganization of a MAS, the desired outcome remains efficient completion of the system objective. In 9Matson and Deloach, 2003), a system of reorganization based on dynamic capability evaluation is presented and discussed next.

In general, Matson and DeLoach's approach to reorganization of MAS first involves the evaluation of the system's ability to perform a desired task. Based on this evaluation, agents may decide to either proceed to satisfy the organizational goals, relax some goals, or abandon the process of reorganization and task acceptance altogether. The foundation of this approach is an organizational model consisting of goals, roles, agents, and capabilities. Based on this model, certain evaluative constraints are applied to the process. First, there must exist knowledge of which agents are available for inclusion in the system. Second, it must be determined what necessary capabilities exist in order to satisfy the demands of a role. Third, an assessment of the capabilities of all available agents must be made to determine their respective qualifications for acceptance of a given role. To perform this step, the authors have devised a capability taxonomy rooted at the abstract level. Leaf nodes of this taxonomy represent concrete functions and capabilities of an agent, such as the types of sensors (sonar, infrared, etc.) and motivators (wheels, tracks, etc.) the agent is equipped with. Finally, limitations applied to roles must be taken into consideration.

2.2 Six Step Evolution Process

Considering these constraints, Matson and DeLoach formulated a six step evaluation process, which begins with the broad definition of system goals. Following this, the broad goals must be reduced into a simpler, structured format. Using this structured form of the system goals, the process determines all roles which will be required to complete the prescribed objectives. Using the knowledge of available agents, a definition of the capabilities of each available agent is collected and used to assign roles to agents capable of successfully assuming the roles. Following this evaluation process, agents can determine whether or not they can satisfy the global system goal, or whether they should relax some goal constraints or abandon the goal altogether. If it is determined that all available agents can perform the necessary roles sufficiently, organization can then take place.

Several such methods exist today. One "general purpose" method has been developed and discussed in (.Zhong and DeLoach, 2006). This method, based on the Organizational Model for Adaptive Computational Systems (OMACS) platform, has been shown to result in optimal network configurations. Another method for performing dynamic organization and reorganization is based on the principle of referral networks. (Yolum and M. Singh, 2003) describes such networks. In referral networks, agents make and sever connections with other agents in the system through the analysis of referrals provided by neighboring agents. An agent wishing to enter the network or alter its set of interconnections once within the network accepts referrals from surrounding agents. From these referrals, agents can form opinions regarding the quality of service provided by other agents and their respective trustworthiness. In (Yolum and M. Singh, 2003), these are referred to as an agent's expertise and sociability.

Agents in referral networks use their knowledge of the trustworthiness and sociability of other agents in the system to decide which agents with which to sever communications or with which to add communication links. Agents that possess high trustworthiness and sociability rankings attract more agents. As the highly trusted and sociable agent gains connections with other agents, its degree increases and it therefore has a greater chance of being referred to other agents in the system. This, in turn, leads to a clustering of agents around the ones seen as most fit, assuming fitness relates directly to trustworthiness and sociability. Clustering of this kind is linked closely to the concept of preferential attachment (Albert and A. Barabasi, 2002) and has been identified in many real-world networks, especially in the internet.

Referral networks are classical and useful, but other methods of reorganization exist. Some of these methods attempt to model biological and chemical organization methods. One such method, related to the concept of stigmergy, is referred to as the digital hormone model (DHM) (Shen, et. al., 2002). This model is based on the understanding that hormonal signals are used often in nature to form organizations of high complexity. In the digital hormone model (as it relates to agents or robots), agents emit activator or inhibitor signals, i.e. hormones, into their surroundings. Once diffused into neighboring agent regions, the agents in these regions combine the incoming hormone strengths with those already present in their area and adapt their behavior based on these recalculated hormone strengths. The actual reorganization process in the DHM requires four steps, which are repeated continuously, assumedly until some goal has been reached. These steps begin with agents assuming roles based on their abilities and associated rules which govern their behavior. Next, execution of roles takes place. This is followed by each agent transmitting and receiving digital hormones to and from their surrounding areas. The final step in this process involves updating each agent's view of the concentration of hormones in its surrounding area.

Some considerations to keep in mind when selecting an adaptation mechanism or creating a new one are the learning rate, stability, and global structure of the MAS formed by the mechanism (Gaston and M. desJardins, 2005). Other considerations are given in (Gaston, et. al., 2004). In this, the authors present the question of which agents should be allowed to adapt in the event of a failure. Three possibilities are considered and include random agent sets, a single agent in the event of a team failure, and all neighboring agents in the event of a single node failure. Furthermore, the authors propose a candidate pool of available agents with which an adapting agent may establish a connection. These are limited to the set of all agents, ex-teammates of the adapting agent, or referred agents, as are used in referral networks. (Gaston, et. al., 2004) continues with this work by outlining a process by which agents adapt given the above noted constraints. The process begins with the construction of the candidate pool and proceeds with several filtering stages. Structural filtering and skill filtering are performed first, followed by degree filtering in which only candidates with the single highest degree are left to connect with the adapting agent.

2.3 Dynamic Reorganization Concepts

(Dignum, et.al., 2004) provides a general overview of dynamic reorganization concepts and examines two metrics useful in examining MAS performance; society utility and agent utility. Society utility is further decomposed into the success of interactions, roles, and structures in the system. Agent utility is not clearly defined, as it differs from agent to agent in heterogeneous agent systems. In addition to these utility metrics, (Dignum, et.al., 2004) classifies several types of reorganization "maneuvers." The first of these, preemptive reorganization, is a viable option in unpredictable environments where possible, or likely, events can be prepared for in order to take full advantage of them. Protective reorganization attempts not to take advantage of possible future events, but instead works to limit the negative effects of such events on the system. Exploitive reorganization takes place after the fact, and seeks to benefit from events that have already taken place. Finally, corrective reorganization attempts to lessen the damage caused by events which have previously occurred in order to maintain system usefulness. Specific methods for performing adaptation are not present in (Dignum, et.al., 2004), but it provides many useful ideas for developing new methods or for elaborating on existing methods.

CHAPTER 3

APPROACH

Communities of agents form virtual organizations that are key components of modern workforce. Dynamic work environments require dynamic organizational configuration. We focus on reorganization both as a rapid reasoning tool and a methodology for a human operator to affect the process of organizational configuration change. In this report we outline a computational framework for agent organizations and tenets of reorganization. We highlight design of a simulated environment as a test bed to showcase our concepts for reorganization. The report concludes with near future plans to complete this research project.

3.1 Foundational Organizational Concepts

In this chapter we will define salient attributes for a computational organization.

We will begin by defining a set of parameters that characterize an organization. First, we will define capabilities.

Definition 1: A *capability* is basic agent ability with a degree in the range from 0.0 to 1.0. We will denote degree of a capability c with D(c).

We assume that there is no decay in capability and agents can only increase their capability. Furthermore, we assume capabilities are mutually exclusive. Let C denote a set of capabilities, which are required in the system for performing all tasks. I.e., $C = \{c_1, c_2, \ldots, c_n\}$. C is the set of all capabilities known by all agents. Each agent will possess each capability c_i to a different degree and may improve it by learning. This provides us with an n dimensional space of capabilities. Let's call this a C-space (Hexmoor and Pasupuleti, 2003). Next, we will define roles.

Definition 2: An action *role,* denoted with r, is a point in C-space that specifies a minimal capability profile to qualify an agent for the role.

For example, with two capabilities c1 and c2, <0.1, 0.5> is a role that an agent may adopt if it's capabilities c1 and c2 exceed 0.1 and 0,5 respectively (Rahman and Hexmoor, 2004). We model an action role with a set of simple rules. For example, a forward role in the game of soccer can be simply modeled with rules resembling the following.

1. If have ball, goal clear, then shoot to goal.

2. If have ball, goal blocked, then dribble

3. If have ball, a teammate is better positioned, then pass ball to teammate.

To execute an action role, an agent senses its environment, picks the best rule to determine an action, and performs the action. Success or failure of actions performed are determined in the environment and not known apriori. At best, an agent may determine a probability of success based on its role fitness to perform it. We'll call the rate of an agent's success its productivity.

Productivity is one of three components for an agent to determine its utility with respect to a goal. The second component is its synergy with others in that role. The third component is the level of fitness of an agent to the role. Fitness of an agent is the sum of ratios of its capability over required capabilities. We are now ready to define an agent's utility with respect to a role.

Productivity = preference (A) * fitness (A, R).

Definition 3: The *utility* of an agent A, performing in a role R_i, denoted by $u(A, R_i)$, is a linear combination of its *productivity,* its *synergy* levels for that role, and its fitness, i.e., $u(A, R) = P(A, R_i) + [(1/\text{sizeof}(R_i)) * \sum s(i, j)]$.

Utility as defined here is intended to denote the relative satisfaction of an agent with a role. If this value is sufficiently high, an agent will be content with its current role. However, if this utility value is low, the agent will be inclined to change its role. As we will discuss norms later in this report, using individual utility as a basis for role change assumes a *selfish norm*. In general, norms are prescribed by a user, which in turn will affect agent decision making. So far, we've considered role changes motivated by utilities alone. However, a major intuitive motivator for role exchange is *opportunity*. In general, opportunity is determined by analyzing environmental attributes that suggest the degree to which adoption of a role by an agent will contribute to system or individual productivities. As an example, a midfielder may see the ball near the opponent's goal and determine that it has a good chance of scoring if it played a forward. Real world computations like this are very rapid and continuous, (Gladwell 2002). Opportunistic computing has also entered technical domains (Marco, et. al., 2007).

Agents need to continually, mentally quantify potential margins of system or individual productivity gains against all possible roles they could adopt. Whichever role candidate will yield the highest marginal gain will be the next role the agent will wish to take. The agent's choice of next role is a proposal that need to be presented to the organization and once processed; the agent may proceed with adoption of it.

For a vacant role, two temporal constraints of the role augment the notion of opportunity. The first is the immediacy of the need to occupy the role. Each open role will specify the urgency of the role. For instance, the team captain will assign a temporal urgency for each vacant role to be filled. Agents who vie for a vacant position, must meet the urgency constraint. Agents must dynamically compute their capability to transition into an open role. Naturally, this capability differs from an agent's innate abilities to perform action. It is not a personality trait. Also, it is not a universal agility or flexibility to take on roles in general. This capability differs with respect to each role and depends on the environmental circumstances. The second temporal constraint is the duration after which the role becomes obsolete and there will not be a need to occupy it. If the validity time window ends, we say such a role is expired. Agents must account for this constraint and should only consider a role if it is not yet expired.

In contrast to an action role, a role that is decision oriented is a manager role. An example of a manager role is captain of a football team.

Definition 4: *Rank* of a role r_i assigns a number to the role that reflects its relative importance in an institution. This is denoted by Rank(r_i).

This is a highly simplified model of a role's valuation in an institution. Using this we introduce a notion of role order. The function "Rank" may return any natural number. The smaller the number the more preferred the role. Role r_j is the most preferred rank if $Rank(r_j) = 1$. Importance of a role is inversely proportional to its rank.

Definition 5: *Role Ordering* (RO) is an ordering of action roles. Each role is assigned a unique rank. I.e., $<Rank (r_1) , Rank (r_2) , \ldots , Rank (r_n)>$ specifies role ordering where $Rank(r_i)$ is the i^{th} position is the rank of i^{th} role.

If $Rank(r_i)< Rank(r_j)$ then role corresponding r_i is preferred over the role r_j that has a smaller rank. RO sets up a trajectory in C-space.

Definition 6: A norm is a convention in the form of a rule shared by all individuals. It may govern role adoption with a set of rules. We denoted the set of norms by N.

For simplicity, we consider organizational norms to be mutually exclusive and non-overlapping. In this paper we will limit norms to rules that govern role change. An example of a norm that governs roles is that individuals incrementally improve their capabilities over time and are allowed to apply for a higher rank in the organization-- this is the *promotion norm*. Another norm that will govern roles is based on utilities. The *selfish* norm will only consider individual utilities whereas *beneficent* will only account for the society's benefits. We assume that norms are determined outside organizations. In our framework, Moe issues a norm.

Definition 7: A *department*, denoted by D_i, is a fixed number of agents who are performing the same role R_i. Each department will require a minimum number of individuals needed to occupy roles in that department at a given time. We assume that departments are static and will not change over time. If there is a significant change for a department such as the number of individuals required, a new organization is formed.

Since departments and roles have a one to one correspondence, ranking one prescribes an ordering on the other. We are now ready to define an organization.

Definition 8: An Organization is modeled as <C, R, D, N>, which is capabilities, roles, departments, and norms.

When an organization is initially populated by individuals, each agent adopts a role depending on its capability level. Each agent will aspire to occupy a role that is highest ranked in the organization.

After an organization is populated, it will experience reorganization where self-motivated individuals may change roles based on the promotion norm. This is shown in Figure 1.

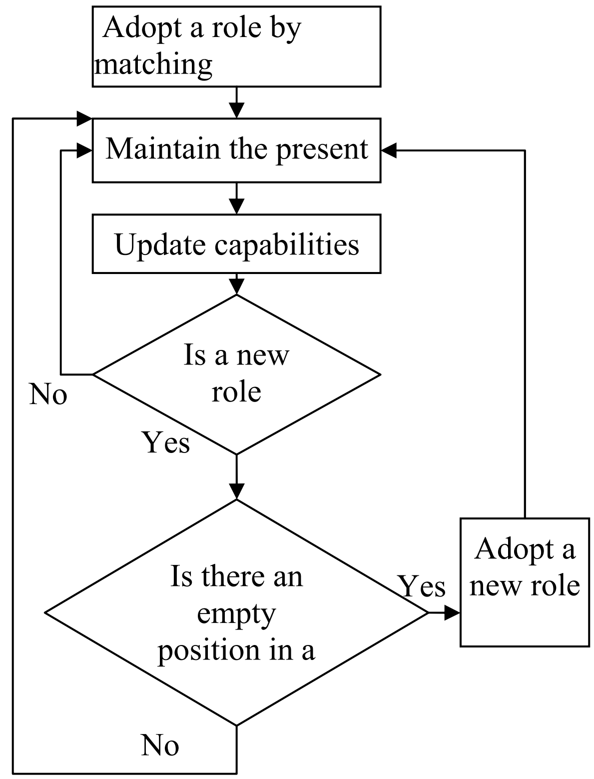

Figure 1: Flow chart for self-initiated promotion norm

We assume that individuals dynamically change roles and move to different departments. To capture a snapshot of organizational configuration we define a *state* that will be defined later.

Definition 9: At any given time, the configuration of individuals in departments of an organization is the state of that department.

States change as often as individuals change their roles. Therefore, there can be many state transitions over time. State transitions are most significantly affected by a norm change.

Thus far, our model of an organization lacks characteristic concepts of agents that will occupy it. In the next section we introduce these notions.

We assume characteristics of agents that will occupy an organization are independent of the organization. Groups of agents have social ties prior to joining an organization that will ebb and flow. Generally, these social ties are independent of the organization and may change little by joining an organization. However, a component of these social ties is synergy among agents that will affect the organization. Next, we define a synergy network keeping in mind that synergy ties are a subset of social ties.

Definition 9: A *synergy network* among a group of agents is a graph among agents where the arcs represent a real value between -1.0 and 1.0 indicating negative or positive influence between pairs of agents. We will use $s(i,j)$ as an directed function that returns the synergy value between agents i and j. Synergy is not reflexive and not symmetric.

Synergy between an agent A and another agent B is a degree of influence from B towards A. This can be positive or negative. We assume authority as a form of synergy. For example a manager role like a captain will have synergy towards players. Synergies change over time. One form of change is experience. Individuals in a department who interact with one another generally develop positive synergies towards one another that is proportional to the duration of time they remain the same department. However, this is not universally true and synergy change goes beyond departmental boundaries. What is important is how one agent's action enables another to perform its action.

Simply modeled, frequency of interactions between agents A and B; that leads B to execute its role increases synergy from A to B. If B is prohibited from executing its action due to A's action, then synergy decreases from A to B.

These changes in synergy cannot be simply determined from the organizational configuration alone. To recapitulate, Synergy changes are dynamic and dependent on interactions.

Beyond individual productiveness, an organization may produce something that is an emergent property and it is not attributed to a single individual. An example is scoring points that are a result of team work. Let's call these system productivity. Similar to an individual, we model a system utility that is a measure of the relative satisfaction of the organization.

Definition 10: *Organizational utlilty* is the sum of individual productivities plus balance of synergies among individuals in the organization.

Definition 11: A state of an organization is the combination of current, active norm and profile of roles occupying departments of the organization, i.e., composition of roles.

Since there are many possible role compositions and there are several possible norms, an organization can be in numerous states. A change in organizational role composition or a change in active norm will yield a state change. States of the system are completely observable and changes in states are under our control. Since state changes are not dependent on the sequences of past states, we can make the Markovian assumption. The nature of organizational state space fits a Markov decision process MDP. There are well known methods for solving MDPs such as the value iteration method (Aicardi, et. al., 1987).

Simulation testbed will be described in chapter 5.

3.2 Netlogo implementation of Soccer Test bed

In order to illustrate the implementation of reorganization, we have designed a model that will be implemented using Netlogo. Netlogo is a java based cross-platform multi-agent programmable modeling environment for simulating natural and social phenomenon. It is freely available online at http://ccl.northwestern.edu/netlogo/.

This platform is particularly well suited for modeling complex, dynamic systems. Experimenters may instruct hundreds or thousands of entities (dubbed "agents") that are all operating independently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals.

The Domain which we will use is a simplified version of the game of Soccer (see "http://en.wikipedia.org/wiki/Football_(soccer)". In order to effectively demonstrate our reorganization principles, requirements we used to select a domain for effective implementation are the following:

i. Availability of a set of explicit rules. Soccer is a classic sport with official rule sets.

ii. Opportunity to rapidly implement a variety of organization types. Soccer formations and defensive and offensive posturing yield a broad spectrum of organizational variety.

iii. A dynamic environment. Soccer as a team sport provides a very dynamic environment.

iv. Possibilities for role exchanges between agents. Players in the game of soccer flexibly move between roles

v. Ability to quantify and monitor performance for agents individually as well as an organizational unit.

vi. Criteria for determining fitness of each individual (i.e., agent) against particular roles.

vii. Opportunity to monitor the performance of a particular agent that possess dynamic changes in the environment and organization

We selected the Soccer game as a domain in order to demonstrate our ideas. It provides the qualities we outlined for a domain. For simplicity, we will neglect a few rules in the game such as "off side", "fouls" and "Scenarios where the ball moves out of the playing field", "penalty shots", "corners", etc. These do not make considerable contribution towards demonstrating the proposed techniques.

The simulated model has two teams playing soccer. The user is provided with a "Norm Selector" input in order to direct the players in the kind of formation they should play to achieve the game mission. Moe can observe the system dynamics in a graph that displays changes in utility for the system with respect to time. Simultaneously, Moe can choose to select a particular agent and monitor her performance in a particular role, which is also represented by the graph denoted “Productivity wrt Time”. Moe may compare the productivity of an agent in different roles.

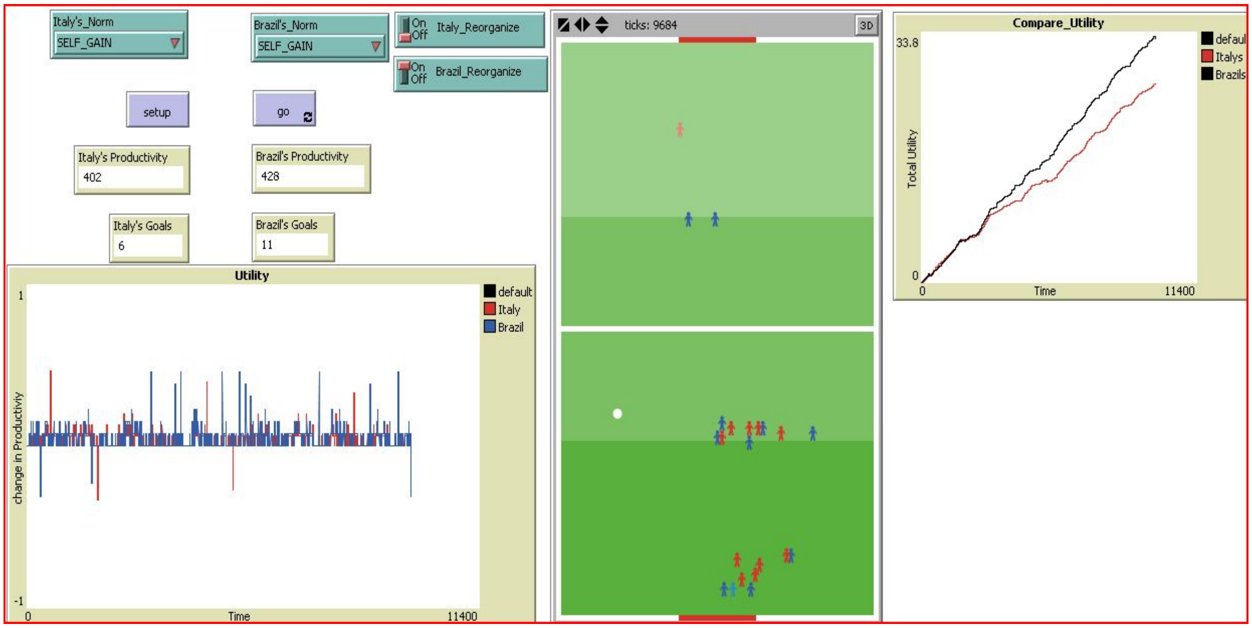

Figure 2: A typical screenshot

Figure 1 is a sample screen and shows various controls for Moe to monitor the system dynamics. The explanation for buttons and menu bars is given next.

1. SETUP: This button is similar to the power button to the system. When this button is invoked the system interprets the code and draws the layout on the screen and the system is ready to run.
2. GO: This is similar to “Run” command. It starts the actual simulation of the program. Each instance is a clock cycle.

3. NormSelector A/B: The drop down menu bars on the left of the screen allows the user to decide on the “mode” or “organization” type to be adopted by the players while playing. She may choose from a preset range of norms.
4. GoalsA/B: These dialog boxes show the number of goals scored in the game by either of the teams respectively.
5. Utility: It is graph drawn to represent the net effective utility of the system, in the context of selected norm. The graph of utility represents “utility wrt time”.

CHAPTER 4

IMPLEMENTATION

The simulation is implemented in NETLOGO, the environment is has two three entities.

- World
- Agents
- Time

World: World is the environment in which the players play the actual game. It is laid out with a scaled down dimensions of a real world soccer field. It has borders on all sides. The agents cannot cross the borders of the field.

Agents: Agents are the players of the soccer game in our case. Each player has his own set of variables and he respects the global rules of the game. The agents are divided into two teams namely "Team Italy" and "Team Brazil".

Time: It is measured by the clock cycles with which the simulation progresses.

4.1 Soccer Simulation:

The game of soccer has eleven players on each side. One player known as Goal keeper, is assigned to protect the goal post and prevent the opponents from scoring goals. The team with maximum number of goals at the end of the time wins. In our simulation we modified the game by ignoring some rules which are not essential for our purpose.

The game in our simulation is designed in the following fashion.

4.2 Capabilities: In this simulation we assume each player has a set of capabilities associated with him, which may or may not be unique. The capabilities are variables that are subject to change with the progress of the game. The set of capabilities each player has are

- Speed
- Accuracy with handling the ball
- Ability to kick the ball (far/near)

Speed**:** It is the variable which represents the speed with which a player moves across a field. His ability to chase the ball move the ball with him. The range of speed is between 1-10, where 1 is the minimum speed and 10 being the maximum speed with which he can move. The ability to be fast is required in one role and is not essential in another role.

Accuracy: This variable sets the players accuracy in handling the ball in the field . His ability to tackle with the ball when the ball is within his reach is decided by this parameter. Accuracy of a player ranges between 20^0-100^0. Imagine a player as the center of an arc and the perimeter if the arc is the range in which the player can move the ball along with him and get from the opponent. A player with large accuracy angle has ability to move the ball around him more and has more possibility of obtaining the ball from the opponent. So higher the angle, higher is his ability to obtain the ball.

Kick: This ability of kicking a ball decides the distance a ball moves when a player kicks it. a player approaches the ball , when the ball is within his reach he tries to kick the ball towards the goal post. The distance the ball moves across the field is decided by the value of the "kick" variable the player of the ball has.

We move on to the roles part of the simulation, the game is divided into a set of roles, where each player can play only one role at a time.

4.3 Roles: There are four roles a player can possibly play in this simulation. A player assuming any of the roles can perform only to the extent the role allows him to do so. For example a player can move at relatively high speed 9. But he assumed a role which allows him to perform only up to a speed of 6, now he can only perform 6 at his best. He cannot cross the role ceiling. In the alternate case when a role requires him to perform at a speed of 5 , but the players capability is only 4, in this case he can perform only at his maximum speed, thus making the player a bad choice for the given role.

- Forward
- Defender
- Mid-fielder
- Goal Keeper

Forward: This role involves a player playing at the forefront of the field, ie; nearest to the goal post. We assumed, at this area of a play the player has to have high speed, moderate accuracy and good kick. We assume these qualities as a requirement for a player to be in this role.

Defender: This role involves preventing the opponent team to score a goal. This role's field of action is near one's own goal post. The player's job is to prevent the opponent from nearing the goal post and reducing his possibility of scoring a goal. The requirements we assumed for this role are high accuracy, low/moderate speed, short kick.

Mid-Fielder: The role of is at the middle of the field. These players are the ones who start the game. Their main objective is to get the ball into their control and help their team to score a goal. These players are responsible for most of the extreme movements of the ball. The requirement of a this role is

High accuracy, moderate speed, and medium kick.

Goal keeper: Although the position of goal keeper is only a single position, we consider it a department because of the uniqueness of the capabilities involved in the role. The goal keeper role has some requirements. The role of a goal keeper requires a player to possess high accuracy, high speed and moderate/low kick.

Departments: In this simulation of soccer game, we implemented in the following manner:

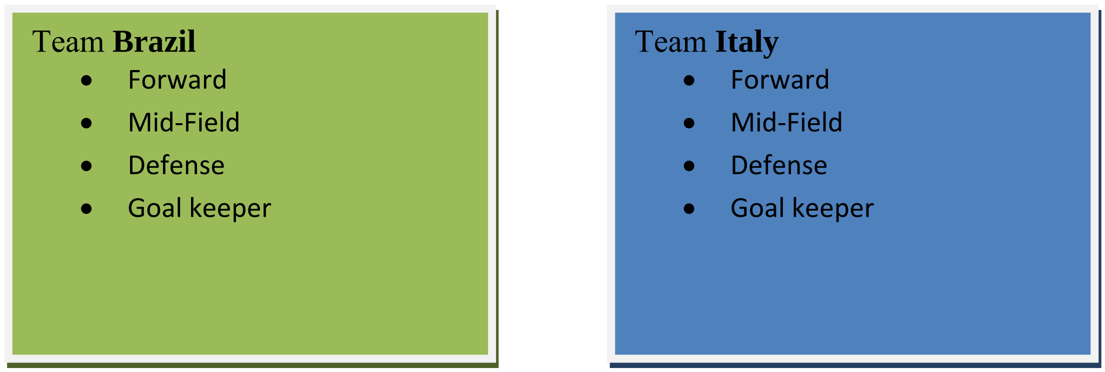

Figure 3: Simulation of Soccer Game

Each Department has a set of players playing the required roles of the department

Forward: All the roles in this department are forward roles and all the players in this department are required to play the role of a forward player.

Mid-Field: All the roles in this department are mid-field roles and all the players in this department are required to play the role of mid-fielder. There are normally four players in this department

Defense: All the roles in this department are mid-fielders and and players play as mid-fielders. There are three players in this department

Goal-Keeper: The player in this department is expected to play the role of a goal keeper. Though this department has only one role, we consider it as a separate department due to the uniqueness of the role.

4.4 Norm: We implemented norm in this simulation. The norm in this game acts as the motivation or attitude with which the players play the game

We designed three norms, each of it differs from the others in unique way.

The norms implemented are Attack, Defend, Self-Gain. In the game the end user or MOTL gives the norm which governs the game.

Norms and reorganization:

Attack Norm: In this norm the players are expected to play with attack mode on their mind. The game is modifies such a way that there are more number of players in the forward department, four instead of three. With more number of players in the forward the game tends to be more aggressive at the cost of weakening the mid-field and defense departments. This is more offensive approach. The reorganization among the players is mainly motivated by the attack norm. The game starts normally and after a period of 500 clock cycles the system checks for any other player who has more capabilities to play in the forward. If it does not implicitly find one it calls for a reorganization and reassigns the roles to the players. The system does not look for best or worst performance it just does the reorganization until Moe is satisfied with it. Only Moe can instruct a system to stop reorganization. So Moe is the one who controls the extent to which she can permit reorganization.

Defend Norm: In this norm the players are expected to play in a defensive mode. The defense department has four players instead of regular three. With more players in the defense the game tends to be more defensive at the cost of losing aggression ie; by weakening the mid-field we also reduce the support to the forward department. After a period of 500 clock cycles; the system checks for players who can be more effective to play in the defense department, and calls for a reorganization. The system this way keeps reorganizing until it receives any instruction from Moe.

Selfish Norm: With this norm players play in the regular department layout configurations. Instead of caring for the team, they play to maximize their productivity and improve their utility. When reorganization is initiated, the system checks if a player is suitable for the role she is playing. If the system considers the player can perform better in another role it calls for a reorganization. This is performed by computing the average productivity of the department. If the player falls below the department average she is underperforming so she is replaced. The system maintains reorganizing until Moe aborts it.

4.5 Synergy: In this simulation, we take the average of the departments average productivity and compare it with the individual synergy. If players productivity is more than the departments productivity we assume she has positive synergy. If she is performing below his capability we assume that she has negative synergy. Each player in the game has a synergy with respective to their department.

CHAPTER 5

RESULTS

We consider three cases to present the results of the simulation

5.1. Case 1

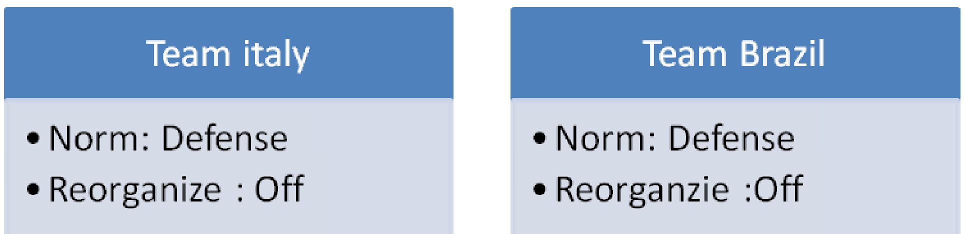

Figure 4: Simulation of Results, Case i

In the Case I, we allow the simulation run for a time period of 11400 clock cycles. The

figure 6.3

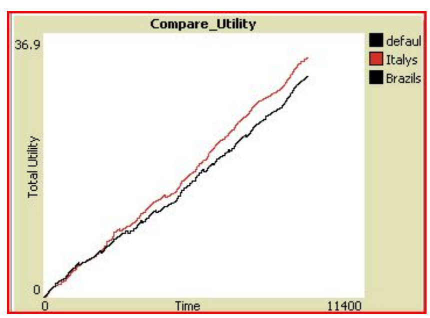

Figure 5: Simulation run for a time period, Case i

The resultant graph between the respective team utility and the time .In this case both the teams norms are set to “Defense”. There are four players in the defense department and three players in the mid-field and three in the forward department. The graph indicates that with the progress of the game the productivity of each team is increasing and as a result of this the utility of each team is increasing. At the point of taking this screen shot ie; at 11400 clock cycles. The utility of both Team Italy and Team Brazil is continuously increasing. With the reorganization switch Off for both of them, there is no reorganization performed. As the graph shows there is only a very little difference between the utilities of both teams. The utility of Team Italy is 36.9 and the utility of Team Brazil is 35.1. The difference between the utilities of both the teams is 36.9-35.1 which equals to 1.7 units. The players of each team come with a set parameters at the beginning of the game, so this simulation shows that the game has been slightly in favour of Team Italy. Since there is no reorganization in this case there is no effect of it.

5.2 Case ii

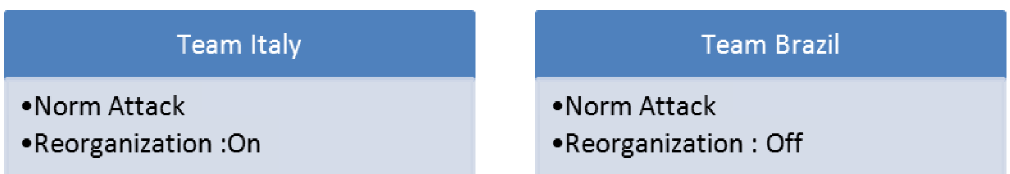

Figure 6: Simulation of Results, Case ii

In this case we set the norm of both the teams in attack mode. And the reorganization switch is turned on for Team Italy, and it is turned to off for Team Brazil. There are three players in Defense and Mid-Field department and there are four players playing on the forward department. The resultant graph of utilities of both teams at a time of 11400 clock cycles is shown in figure 6

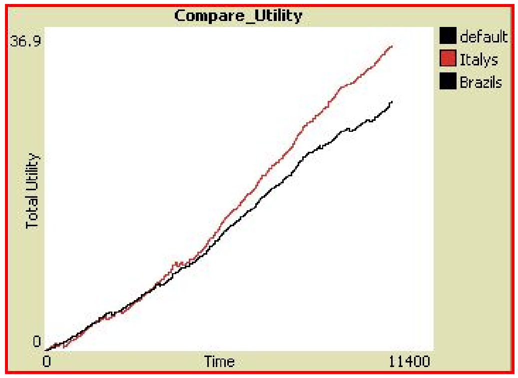

Figure 7: Simulation run for a time period, Case ii

At the time when the readings are noted the graph shows, Team Italy's utility 36.9 and Team Brazil's utility at 31.2. Due to the implementation of the reorganization algorithm the Team Italy has a considerable advantage over the opponents. The difference between the utilities of both the teams is 36.9-31.2 which equals to 5.7 units of utility. With progression of the game the utilities tend to increase as the reorganization algorithm deploys best guy for the best possible job to obey the norm of attack until it reaches a maximum value where it settles. In this case we were successfully able to show the effectiveness of the reorganization algorithm.

5.3 Case iii

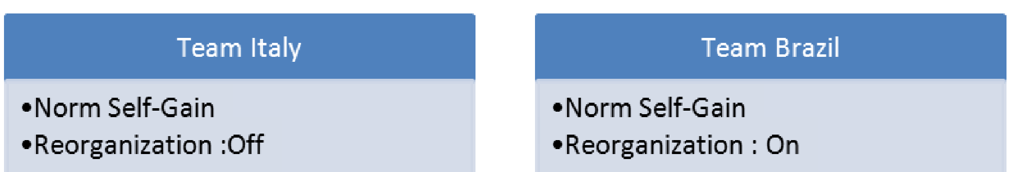

Figure 8: Simulation of Results, Case iii

In this case we changed the norm to Self-Gain or Selfish norm for both the teams. And the reorganization switch is turned on for Brazil and it is set to off for Italy. In Selfish norm the game layout stays to standard size of the departments with three players in forward and defense and four players in mid-field. With this norm governing the play each player tries to maximize the his own utility and productivity even at the cost of overall drop in the team productivity in some cases.

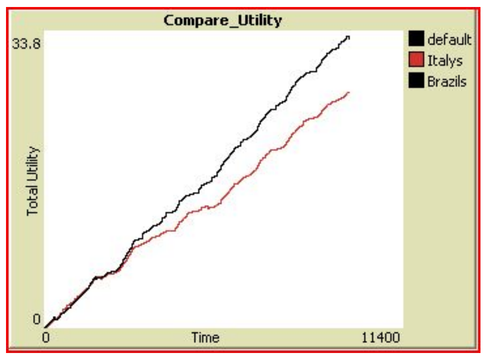

Figure 9: Simulation run for a time period, Case iii

The graph in figure 6.7 shows the utility of each team at the time of 11400 clock cycles. The utility of team Brazil is considerably higher at 33.8 and the utility of Team Italy is 27.3 both of the utilities differ by 6.5 units of utility. This result also clearly shows the advantage of using the reorganization in our norm. We witness an overall drop of the team utility at 33.8 being the maximum for the same amount of time as the earlier cases. This is due to the norm which stresses over individual utility over the overall utility of the team. This may not be always true because when we consider a case where all the players in the team are exceptionally talented and are most suitable for their roles , a condition of utopia then the self gain norm may possibly enhance the overall utility of the team.

CHAPTER 6

CONCLUSION

We have made novel strides in managing massive agent organizations that contributes to development of our man on the loop paradigm. Our methodology allows for Moe to prescribe normative patterns of behavior, which in turn guide the reorganization process. Individuals most fit to play their current roles stay while others are directed to surroundings that augment their synergy. We have demonstrated with the popular game of simulated soccer but the results are generic and transfer to other domains. The results demonstrated our design methodology of reorganization of agents in a multi agent simulation.

6.1 Future plans

This research can be further extended in many folds. Like organizations of the *agents* within the agent society can be blended on the top of cultural parameters which will reveal the full spectrum of PDI as difference in power will exist. Moreover this hierarchical distribution will allow us identifying the key agents which will negotiate with *MOTL* instead of every individual group member. These key agents with higher power can be treated as leaders which can also supervise the subordinate agents. Using this concept over *MOTL* we can also develop Agent Reorganization based on their cultures rather than just the capability, requirement. The concept of using Culture as a norm for agents to reorganize is to itself another potential area to be explored.

Since we can model operations of an organizational by an MDP, we can explore automatic solutions to organizational changes as parametric state transitions in MDP.

REFERENCES

[1] M. Aicardi, F. Davoli, and R. Minciardi. Decentralized optimal control of markov chains with a common past information set. IEEE Transactions on Automatic Control, AC-32:1028–1031, 1987.

[2] D.S. Alberts, and R.E. Hayes (2006). Understanding command and control. Command and Control Research Program (CCRP), available online at: www.dodccrp.org.

[3] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, 2007. Opportunistic computing for wireless sensor networks, Proc. IEEE mobile ad hoc networking conference.

[4] K.M. Chang (2005). The performance of edge organizations in a collaborative task. (Doctoral dissertation). The Naval Postgraduate School, Monterey, CA.

[5] H. Hexmoor and S. Pasupuletti, 2003. Institutional versus Interpersonal Influences on Role Adoption, In AAMAS workshop titled: Representation and approaches for time-critical resource/role/task allocation, J. Modi and T. Wagner (Eds), Melbourne, Australia.

[6] M. Gladwell, 2002. The Tipping Point: How Little Things Can Make a Big Difference, Back Bay Books.

[7] E. Matson and S. DeLoach. Using Dynamic Capability Evaluation to Organize a Team of Cooperative, Autonomous Robots. Proceedings of the 2003 International Conference on Artificial Intelligence (IC-AI '03), 2003.

[8] A. Rahman, and H. Hexmoor, 2004. Negotiation to improve Role Adoption in Organizations, In Proceedings of International Conference on Artificial Intelligence (IC-AI), Pages 476-480, CSREA Press.

[9] Albert.R, & Barabasi.A. (2002). Statistical Mechanics of Complex Networks. In Review of Modern Physics (p. 74).

[10] Barber, K., & Martin, C. (2001). Dynamic Reorganization of decision making groups. Proceedings of 5th autonomous agents .

[11] Carley, K., & Gasser.L. (1999). Computational organizational theory . Multiagent systems : A modern approach to distributed artificial intelligence , 299-300.

[12] Chaimowicz, L., Campos, M., & Kumar, R. (2002). Dynamic Role Assignment for Cooperative Robots . IEEE Conference on Robotics and Automation .

[13] Dignum, V. (2004). A Model for organizational interactions based on Agents, founded in Logic, SIKS Dissertion Series 2004-1. 2004: Utrecht University ,.

[14] Dignum, V., Furtado, V., Dignum, F., & Melo, A. (2005). Towards a Simulation Tool for Evaluating Dynamic Reorganization of Agent Societies.

[15] Gatson, M., & desJardins, M. (2005). Agent Organized Networks for Dynamic Team Formation. AAMAS .

[16] Gatson, M., Simmons, J., & desJardins. (2004). Adapting Network Structure for Efficient Team Formation. AAMAS Workshop on Learning and Evolution in Agent Based Systems .

[17] Glasser, N., & Marignot, P. (1997). The Reorganization of societies of autonomous agents. MAAMAW , 98-111.

[18] Handley, H. .., & Levis., A. (2001). A Model to evaluate the effect of organizational adaption. In H. .. Handley, & A. Levis., Computational and Mathematical Organization Theory 7 (pp. 5-44). kluwer academic publishers.

[19] Hexmoor, H., McLaughlan, B., Tuli, G., (2008). Natural human role in supervising complex control systems, In Journal of Experimental and Theoretical Artificial Intelligence, Taylor and Francis, pp. 59-77.

[20] K.C.Chang, P.E.Lehner, A.H.Levis, A.K.Zaidi, & Z.Zhao. (1994). On causal influence logic. Technical report,George Mason university Centre for Excellence.

[21] Kazakos, P., & Zaidi, A. (2008). An Algorithm for activation timed influence nets. ieee .

[22] Matson, E., & DeLoach, S. (2004). An Organizational Model for Designing Adaptive Multiagent Systems. AAAI Workshop on Agent Organization .

[23] Matson, E., & DeLoach, S. (2003). Using Dynamic Capability Evaluation to Organize a Team of Cooperative Autonomous Robots. Proceedings of the 2003 International Conference on AI .

[24] Shen, C., Choung, C., & Will, P. (2002). Simulating Self-Organization for Multi-Robor Systems. IEEE/RSJ International COnference on Intelligent Robots and Systems .

[25] Valetto, G., Kaise, G., & S.Kc., G. (2001). A moblie agent approach to process-based dynamic adaption of complex software systems. 8th European Workshop on Software process technology, pp. 102-116.

[26] Victor, L., & Abdullah., S. (2007). Multiagent reinforcement Learning and Self-Organization in Network Agent.

[27] W.L.Smith, J. (1996). Influence Net Modeling with causal stregths : An Evolutionary Approach. Proceedings of command and control research technology symposiun. monterey.

[28] Yolum.P, & Singh.M. (2003). Emergent Personalized Communities in Refferal Networks. IJCAI workshop on Intelligent Techniques for Web Personalization .

[29] Zaidi, A. K., & Papatoni-Kazakas, P. (2007). Modelling with Influence Networks Using Influence Constants: A New Approach.

[30] Zhong.C, & DeLoach.S. (2006). An Investigation of Reorganization Algorithms. International Conference on Artifiial Inteligence .

VITA

Graduate School
Southern Illinois University

Aruntej Seelam Date of Birth: January 20, 1986

505 S Poplar St,Apt 7, Carbondale IL -62901

Dno: 1-238/1 Kothapet, Mangalagiri, AP, India – 522503

aruntej.s@gmail.com

Anna University, Chennai
Bachelor of .Engineering, Electronics and Instrumentation Engineering..

Thesis Title:

Reorganization of Massive Multiagent Systems: MOTL/O

Major Professor: Dr. Henry Hexmoor, PhD