Oversight of Reorganization in Massive Multiagent Systems

Henry Hexmoor
Department of Computer Science, Southern Illinois University,
Carbondale, IL 62901, USA
Email: {hexmoor}@cs.siu.edu

Abstract. We have explored mechanisms for converting organizations to an edge type organization. Beyond structural differences, organizations differ in information flow network and information sharing strategies. We review organizational adaptation. A model of computational organization and reorganization is presented using dynamic roles. In addition to self-organization, our model allows human oversight and guided reorganization. This article lays a foundation for automatic organizational adaptation and human supervision. Our model is exemplified with simulated soccer.

Keywords: organization, multiagent systems, supervision

1. Introduction

Hierarchical command and control (C2) structures are ineffective [3]. Power to the Edge (PE) is an information and organization management philosophy that has superseded traditional organizational paradigms for C2 [3]. PE provides empowerment for edge members, as well as superior, interoperable, agile, and shared awareness among all the members in the organization. PE provides adaptability in dynamic situations [8]. Man on the Loop (MOTL) complements PE with a set of computational tools for systematically altering organizational components for agile response to dynamics in large population multiagent systems [16][17][22]. Reorganization is necessary to allow communities of agents that are possibly robots to reconfigure their organizational structure with agile response to changes in the environment.

Communities of agents form virtual organizations that are key components of modern workforce in diverse fields in the military and the business world. 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 paper we outline a computational framework for agent organizations and tenets for rapid reorganization. We highlight an implemented simulated environment serving as a testbed to showcase our concepts for reorganization.

We have explored principled mechanisms for converting a hierarchical organization to an edge type organization. Other than structural differences, organizations differ in information flow networks and information sharing strategies. Many types of organizational adaptation are possible and require in-depth research that we anticipate to continue for our future work. This article lays the foundation for automatic organizational adaptation. We begin by outlining related work and background in section 2. In section 3 we present a virtual organization model with features for rapid reorganization. Section 4 describes an implementation of a simulated testbed that facilitate validation of our salient concepts. Observations and conclusions are provided in sections 5 and 6 respectively.

2. Background

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 forms, dictates the communication interactions among agents as well as the distribution of roles and authority throughout the system. When motivated to adapt, agents should do so with or without human intervention and in a manner that improves the overall performance of the system. Alternatively, adaptation might also be motivated to improve organizational cohesion and increased synergy among individuals. Several motivating factors for reorganization are discussed here as will current proposed methods for performing dynamic reorganization in MAS [5].

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 [11].

The impetus for 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 (i.e., quality-based) [21]. In this work, the authors illustrate three scenarios resulting in a need for reorganization [19][20]. 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 [7], 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 [20], 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.

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 [14][15].

Several such methods exist today. One “general purpose” method has been developed and discussed in [29]. 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; authors in [27] describe 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 [26][27]. 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 [27], 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 [2] and has been identified in many real-world networks, especially in the internet [2].

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) [24]. 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 [11]. Other considerations are given in [12]. 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 [11] and continue with this work by outlining a process by which agents adapt given the above noted constraints [12]. 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.

Dignum provides a general overview of dynamic reorganization concepts and examines two metrics useful in examining MAS performance [9]; 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, several types of reorganization “maneuvers” are classified in [10]. The first of these, pre-emptive 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 [10], but it provides many useful ideas for developing new methods or for elaborating on existing methods [6].

Multiagent reinforcement learning provides protocols for agents to dynamically adapt to changes in their environment, which does not require deep domain knowledge. Agents learn by iterating an initial policy, which will dictate their adoption of roles in an organization. When there is a change in the organization, reinforcement will guide agents to self-organize to adapt to organizational changes and select more appropriate roles. We envision learning to become one of future thrusts. However, it remains outside our current scope.

3. An Organizational Model

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

Definition 1: A *capability* is a basic agent ability with a degree in the range from 0.0 (i.e., the least) to 1.0 (i.e., the most). We will denote degree of a capability type c with D(c). I.e., $D(ci) \in \{0..1\}$.

We assume that there is no decay in an agent's capability and agents only increase their capabilities. In the real world, capability atrophy is a significant factor but our assumption applies to skills that are continually practiced as in an active season for a sport such as football. For simplification, we assume that capabilities are mutually exclusive. Changes in one capability do not affect others. In reality, capabilities are highly interdependent. In fact, capabilities that are similar rely on same basic abilities of the agent. For example, if a player is incapacitated in a certain way, it will diminish all dependent capabilities. For example, a knee injury will affect running as well as all capabilities for ball handling.

Let C denote the superset of capabilities, which are required in the system for performing all tasks. I.e., $C = \{c_1, c_2, \ldots, c_n\}$. C will be common knowledge to all agents. Each agent will possess each capability c_i to a different degree and may improve it on its initial level by learning or practice. This provides us with an n dimensional space of capabilities. Let's call this a C-space. Since it is previously defined elsewhere, we do not repeat or elaborate it further and refer the reader to [17]. Next, we define a class of roles that involve action.

Definition 2: An action *role*, denoted with r, is a point in C-space that specifies a minimum required capability profile to qualify an agent for the role. We will denote the set of all available roles by the set R and |R| is the total number of roles.

Definition 2 provides a quantitative model for action roles. For example, with two capabilities c1 and c2, <0.1, 0.5> is a profile for a role that an agent may adopt if the agent's capabilities for c1 and c2 is at least 0.1 and 0.5 respectively [23].

An alternative for quantitative model is a qualitative characterization where 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 similar to the following three:
1. If ball is in possession and goal is clear, then shoot to goal.
2. If ball is in possession and goal is blocked, then dribble
3. If ball is in possession and a teammate is better positioned, then pass the ball to teammate.

To execute an action role, an agent senses its environment, picks the rule that has the best match to prevailing conditions for determining an action, and performs the prescribed action. Success or failure of actions performed are determined in the environment and are not known a-priori. At best, an agent may determine a probability of success based on its fitness to perform the role. We'll call the actual rate of an agent's success in role performance its productivity. Agents' private preferences for roles are defined next.

Definition 3: A *preference* function for an agent, denoted with preference, specifies a degree of preference for a role range from 0.0 to 1.0. We will denote preference for an agent A for a role r with preference(A, r). I.e.,
$preference(A, r) \in \{0..1\}$.

Determining match between an agent and a role is captured in a function we call fitness.

Definition 4: At a given time, *fitness* of an agent with respect to a role ri is the sum of available capability degrees of that agent at that time over all required capabilities for that role.
I.e., $\mathrm{fitness(A, r)} = \sum i\ \mathrm{D(ci)}$.

Productivity is one of the main components for an agent to determine its utility with respect to a goal. We assume that fitness and preference contribute equally to productivity.

Definition 5: Productivity of an agent at any time is defined as sum of the products of (a) preferences for all agent roles, with (b) fitness levels over respective roles. I.e.,

$$Productivity(A) = \sum r\ (preference(A, r) \times fitness(A, r)) \text{ (Equation1)}$$

Another component for determining agent utility is the agent's level of *synergy* with others in a given role. Synergy between an agent agent A and another agent B is a degree of influence generated from B towards A. We will not elaborate much on the nature of influence but note that influence depends on many attributes between two individuals. This influence can be positive (i.e., corroborative) or negative (i.e., detrimental). We assume authority to be a special form of synergy. For example, a manager role like a team captain will exert synergy towards players in the team. Synergies change over time. One form of change is experience. Individuals in a group who interact with one another generally develop positive synergies towards one another that are proportional to the duration of time they remain the same department. However, this is not universally true and synergy change goes beyond group 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 will lead B to execute its role, increases synergy from A to B [25]. If B is prohibited from executing its action due to A's action, then synergy decreases from A to B [24]. In definition 6, we will define synergy among a group as a synergy network and a directed version between a pair of agents.

Definition 6: 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 [18][28]. We will use *s(i,j)* as a directed function that returns the synergy value between agents i and j. Synergy function is not reflexive and not symmetric.

Changes in synergy cannot be simply determined from the organizational dynamics alone. Synergy changes are dynamic and dependent on interactions. Research on synergy has significant overlaps with research on collaboration and coordination and specifically with the idea of coordination without explicit communication (i.e., that is termed *stigmergy* in robotics). Further elaboration is outside our current focus [26]. Although synergy ties might typically be a subset of social ties, here we keep synergy networks and social networks distinctly apart.

We are now prepared to define an agent's utility with respect to a role.

Definition 7: The *utility* of an agent A, performing in a role ri, denoted by u(A, ri), is a linear combination of its *productivity,* and its *synergy* levels for that role. I.e.,

$$\mathrm{u}(\mathrm{A}, \mathrm{ri}) = \mathrm{productivity}(\mathrm{A}, \mathrm{ri}) + \frac{1}{|\mathrm{ri}|} \times \sum_{\mathrm{i}}^{\mathrm{j}} \mathrm{s}(\mathrm{i}, \mathrm{j}) \text{ (Equation 2)}$$

In equation 2, |ri| is the total number of capabilities required for role ri. Utility as defined here is intended to denote the relative satisfaction an agent experiences with respect to a role. If this value is sufficiently high, the agent will be considered to be content with its current role and lack the desire to vacate the role for another. However, if this utility value is low, the agent will be inclined to change its role to one that may yield a higher utility value.

Using individual utility as a basis for desiring role change assumes the agent is operating with a *selfish norm*. In general, *norms* (presented in our definition 10) are prescribed by an external system oversight user and not a player in the system. Norm selection in turn will affect agent decision making. Thus far, we've considered role changes motivated solely by utilities. 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 future system or individual productivities. As an example, a football midfielder may see the ball near the opponent's goal and determine that it has a good chance for scoring if it played a forward. Real world computations like this are very rapid and continuous [13]. Opportunistic computing has been used in a variety of domains [1]. Blackboard systems is an example of a computational system that has heavily relied on opportunities for developing evolving, distributed solutions [4]. Opportunity is an indirect attribute that cannot be preserved or predicted. As a volatile quantity it can typically be implicitly incorporated in conditional and contingent rules but we will not formalize it beyond a general discussion. For role adoption, we envision rules as in "if a role r is vacant, adopt r".

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 undertake. The agent's choice of next role is a proposal that need to be presented to the organization and once permission is granted, 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 for 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 the action. This is not a personality trait. Also, it is not a universal agility or flexibility to take on roles in general. These capabilities differ with respect to each role and depend 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 has 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. Organizations maintain importance values over roles that we will define in the term *rank*.

Definition 8: *Rank* of a role $r_i \in R$ is a function assigns a number to the role that reflects its relative importance in the organization. This is denoted by Rank(r_i). I.e., $\mathrm{Rank}(ri) \in \{1..|R|\}$

Here, |R| is the total number of roles. Our definition of rank is a highly simplified model of a role valuation in an organization. Using this we introduce a notion of role order. The function Rank will return a natural number between 0 and |R|. 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 9: *Role Ordering* (RO) is a complete ordering over action roles. Each role is assigned a unique rank. I.e., <Rank (r_1) , Rank (r_2) , ….. , Rank (r_n) > specifies role ordering where Rank(r_i) is the i^{th} position is the rank for 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. Agents will aspire to move on from low ranking roles to successively higher ranking roles. Therefore, beyond utility considerations, our agents will favor roles that are ranked higher than their current roles. Ranking and utility are attributes that differently affect an agent's decision making for role adoption. A consideration that can further guide role adoption in the entire organization is to specify a cultural mindset we will rather informally define next.

Definition 10: A *norm* is a convention in the form of rules shared by all individuals in the organization. It governs 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 ranking role in the organization-- this is considered to be 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. For simplicity, we assume that norms do not evolve in organizations. In our framework, the system user deliberately sets the norm. Our motivation for the human level control of norms has been to model a military commander or a team coach who imparts his or her own style of interaction to the organization. As norms are selected, system performances as well as styles of interactions are observed and they might be considered in attempting to prescribe other norms.

Definition 11: A *department*, denoted by D_i, is a fixed number of positions occupied with 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 any 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 12: An Organization is modeled as <C, R, D, N>, which are sets of capabilities, roles, departments, and norms, respectively.

When an organization is initially populated by individuals, each agent adopts a role depending on its capability level. At initial adoption, each agent will aspire to occupy a role that is highest ranked in the organization for which it qualifies. During subsequent changes, agents will abide by norms as it permits them to consider combinations of utilities, ranks, or other attributes.

It is a simple extension to consider utility of an organization to be sum of utilities of all agents fulfilling their roles. Similar to an individual, we model a system utility that is a measure of the relative satisfaction for the organization. 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 this system productivity.

Definition 13: Organizational utility, denoted by *ou* is the sum of utilities of all agents fulfilling their roles. I.e.,

$$ou = \sum_{ai,rj} u(ai,rj)$$

here ai is an agent and rj is ai's current role and the sum is over combinations of pairs of agents and their current roles. The resulting value of ou reflects a measure of organization performance. Organizational utility in definition 13 as the sum of individual utilities lacks an account for balance of synergies among individuals in the organization. Ideally, an organization may need to strive toward maximizing synergy among individuals as well as producing the highest yields from members of the organization. However, we conceive of overlapping nuances and possible complications that need to be further explored before a more complete formulation can be presented. We postpone these explorations for future research.

The oversight person guiding the organization may use organizational utility value for reconsidering the desired norm. However, it is possible that performance might not be the desired determinant. Sometimes, for an organization, styles of interaction or certain levels of synergy are the intended outcomes. Performance and style are a subset of considerations and are mentioned here for illustration of our formulation. Specific and more comprehensive sets of considerations are beyond our current scope.

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 Fig 1 as a flowchart. Instead of a algorithmic illustration we will explain it next for brevity. Each agent initially will consider its capabilities and will adopt a role r that best matches its capabilities. Role r is selected for which (a) the agent capabilities meets the role's minimum required capability degrees, (b) the agent's capabilities exceeds the role r's capability requirements by r%, (c) among all competing roles for which the agent is qualified to adopt, the agent's capability excess (r%) is the largest. The agent will keep role r as its capabilities change and improve as we have assumed for our current consideration. If no new roles appear in the organization, there'll be no need for role change. However, if a new role becomes available for adoption, the agent will re-evaluate its capabilities and if meets and its excess role qualification exceed its qualification level for the current role, it'll consider a changing its role to the new role. Roles might become available for adoption either because they are vacated by other holding those roles or by organizational changes where role reassignments occur. Further discussions of role change are outside our current scope.

Fig. 1.

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. First we define a configuration.

Definition 14: At any given time, the distribution of individuals across departments of an organization is the *configuration* of that department.

Configurations change as often as individuals change their roles. Therefore, there can be many configuration transitions over time, which are most significantly affected by changes in the effective norm. Thus far, our model of an organization lacks characteristic concepts of agents that will occupy it. Next, we introduce these notions.

We assume characteristics of agents that will occupy an organization are independent of the organization. Groups of agents will 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.

Definition 15: A *state* of an organization is the combination of current, active norm and configuration of departments in the organization.

Since there are multiple possible configurations and there are multiple possible norm options available, an organization can be in one of numerous states. A change in organizational role composition or a change in active norm will yield a state change. For oversight, individual and system utilities as well as states of the system are completely observable and changes in states are controllable by a change in the effective norm or the configuration. Since state changes are not dependent on the sequences of past states, we can make the assumption that any changes depend on user specifications of changes in configurations and norm; i.e., the Markovian assumption [1]. The nature of organizational state space fits a Markov decision process MDP. For determining desirability of each state, there are well known methods for solving MDPs such as the value iteration method [1]. This means that our formulation for organizations can be used to decide quality of each state and the user can deliberately take action to control the organization by guiding it to the most desirable states.

4. Implementation

In order to illustrate the implementation of reorganization, we have designed a model that is 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/.

The Domain which we will use is a simplified version of the game of Soccer. Netlogo and soccer are particularly well suited for demonstrating 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.

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. We needed to know well an individual as well as how well the entire organization performs.
- vi. Criteria for determining fitness of each individual (i.e., agent) against particular roles. There is a need for attributes to determine degree of match between individuals and roles.
- vii. Opportunity to monitor the performance of a selected agent that can produce dynamic changes in the environment and the organization. E.g., performance of a goal keeper in a soccer match.

We selected the popular game of Soccer as a domain in order to demonstrate our ideas. It provides the qualities we outlined for a domain. In our simulation we modified the game by ignoring some rules which are not significant for our purpose of demonstrating reorganization For simplicity, we 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 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. The following six rules constitute main game rules of interest:

1. A score occurs if the ball passes under the crossbar, between the goalposts, and the entire ball passes completely over the outside edge of the goal line.
2. At the start of each game, the ball is placed in the center of the field. All players must be in their own half of the field and the opponents of the team taking the kick-off must be at least 10 yards from the ball.
3. If the entire ball passes completely over the outside edge of the touchline (sideline), the restart is a throw-in. The throw-in is taken by the team that did not last touch the ball.
4. If the entire ball passes over the goal line, it was last touched by an attacking player, and a valid goal was not scored, the restart is a goal kick. The ball is placed in the goal area and kicked by a member of the defending team.
5. If the entire ball passes over the goal line, it was last touched by an attacking player, and a valid goal was not scored, the restart is a goal kick. The ball is placed in the goal area and kicked by a member of the defending team.
6. If the entire ball passes over the goal line, it was last touched by a defensive player, and a valid goal was not scored, the restart is a corner kick. The ball is placed in the corner arc and kicked by an attacking player. A goal may be scored directly from a corner-kick.

Fig. 2. A typical screenshot

Fig 2 is a sample screen and shows various controls for the human supervisor to monitor the system dynamics. The explanation for buttons and menu bars is outlined 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. This method instantiates all agents in this game. Each agent/player is instantiated with a set of capabilities. Similarly, each role is instantiated with a set of requirements.
   Initially, each player starts with a particular position or role in the game. The field has been given a shading gradient to show different areas of the field which will act as the region in which each department of players play. The notion of departments is explained in a later part of this section. The ball is placed in the center of the field, the center of the midline to be precise. The whole setup is ready for the players to start playing.

2. **GO:** This is similar to the Run command. It starts the actual simulation of the program. Each instance is a clock cycle. Once the Go button is actuated at the start of each clock cycle the program is executed from the method "go" in the code, which is comparable to m*ain* method in a conventional programming language.

3. **Norm Selectors:** There are two Norm selector controls shown in the screen shot. Each of which is a drop down menu type. Each team has its respective norm selector. The user/MOTL selects the norm in which he desired to simulate the game, before start of the game. *Italy_Norm* and *Brazil_Norm* are the norm selector drop down menus of the respective teams. For our simulation we implemented three norms namely Attack, Defend and Self-Gain. Each of which have their own governing rule and parameters which are discussed further on.

4. **Reorganization Switches:** There are two controls, each operate like a regular ON/OFF toggle switch. These switches are used to start or stop the implementation of the reorganization algorithm in the game. Each team, namely Team Italy and Team Brazil, have their respective switch, which are at the user's disposal. The reason for providing a switch is to give the user a freedom to decide when to start/stop reorganization.

5. **Goal Monitors:** These are simple score board like monitors providing the latest scores in the game.
6. **Productivity Graph**: The graph in extreme left side of the screen provides us the information about the productivities in each team with respect to time. The productivity may increase or decrease depending on the norm selected, players capabilities, and the condition of the game.

7. **Utility Graph:** This graph displays utility of each team with respect to time. Utility in this simulation is computed by the effectiveness of the player in obtaining the ball from the opponents, number of goals scored by the team, and number of times the opponents successfully took the ball away from a team.

8. **Field:** The rectangular box in the center of the playing area (shown in Figure 2), is the actual soccer field in which the play takes place. The field has been divided into five regions. There are two goal posts at each end of the field. The gradient shading distinguishes differences of each part of the field. Consider a Team; the patch near to its home goal post (which is supposed to be protected from opponents) becomes the defense region for the team. The lighter shaded patch at the center of the field, which also has includes centerline becomes mid-field for the team. The farthest patch from the home goal post, which is also nearest to the goal post the team is

supposed score goals into becomes the forward region for the team, the same is replicated for the other team.
The players are initially started at certain region of the field which becomes their designated region. The player plays for the most part in his designated region unless there is an explicit need to fetch the ball or until he is commanded by the user to do so. Effectively, a player plays only in the region which he is assigned and expected to play.

The game in our simulation is designed and implemented with the following detail.

Capabilities: 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. In our game each player from the start of the game maintains his capabilities at the same level. The set of capabilities for each player are composed of:

- Player Speed
- Player accuracy for handling the ball
- Player ability to kick the ball (e.g., far or near)

Speed: This is a variable, which represents the speed with which a player moves across the field. His ability to follow the ball and move the ball (i.e., dribble along). The range of speed we coded is between 1 to10, where 1 is the minimum speed and 10 is the maximum speed with which he can move. The ability to be fast might be required in one role and may not be essential in another role. The player performing in an assigned role may utilize up to the maximum of his capability although his role demand a default amount. The player who has more capability than the demand level for the role can utilize only to the maximum level of demand for the role. He cannot perform better than the role will allow him even though he is capable. For example, if a player has a speed of 6 and he assumes a role which demands him to perform at a speed of 9, he can only perform up to his maximum, which is 6. Similarly, if a player assumes a role which demands a speed of 4, he will only perform up to a speed of 4 even though he is capable of a speed of 6.

Accuracy: This variable sets the players' accuracy for handling the ball in the field. His ability to tackle the ball when the ball is within his reach is decided with this parameter. Accuracy of a player ranges between 20 degrees to 100 degrees. 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 move away from the opponent. A player with large accuracy angle has ability to move the ball around him with more degree of control and has more possibility of obtaining the ball from the opponent. Therefore, the larger the angle, his ability is proportionally higher to obtain the ball. Similar to the speed parameter, the player can only exhibit his accuracy to the extent which the role he assumed allows him to perform and he cannot play more accurately than his maximum even if the role requires him.

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 possesses. Like all other parametric constraints, the player has to limit the extent to which he can kick the ball according to the limitations and demands of the role assumed by him. Next, we focus on 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.

Roles: There are four major roles a player can possibly assume in this simulation. A player assuming any of the roles can perform only to the extent the role allows him to do so. The four roles a player can play are:

1. **Forward**: A player assuming the role of a forward plays in the forefront of the field that is the region farthest from his home goal post and most near to the goal post where he is supposed to score a goal. In our simulation, we gave the role the following rules. The speed of the player has to be relatively high (7) in this case. He can have moderate accuracy and he has to have high kick length of 10 units of distance.
2. **Defender**: A player assuming the role of a defense player plays in nearest region of his home goal post. His primary duty is to prevent the opponent team from scoring a goal and bring the ball nearer to the goal post. He should try to steal the ball from the opponent and send it farther in the opponent's field, that is away from his home goal post. In this simulation we set the requirement of a player playing in this role. He should have moderate accuracy with the ball. He can have a moderate speed of 5 and he should have a high kick length, which has to be 15 units of distance.
3. **Mid-Fielder**: This role is at the middle of the field. These players are the ones who start the game. They are assigned to the middle of the field near the center lane. Their main objective is to pass the ball to the players playing in the forward region and also to prevent the opponent from charging further into the defense region. The roles of mid-fielder are pivotal to controling the movement of the ball as their performance decides which side of the field the ball will move. Their support is important for the player performing in the forward roles to score goals as these players are responsible for most of the extreme movements of the ball. The requirement for this role is high accuracy with the ball, they can have a moderate speed of 5 and they should be able to kick the ball at a moderate distance of at least 9 units of distance in the field.
4. **Goal keeper**: There is only one role of goal keeper for each team. Although there are no limitations to the number of players assuming other roles, soccer game has strict rules of only one player playing the goal keeper at any one time. The main duty of the goal keeper is to protect he home goal post by preventing the ball from entering the goal post and to prevent the opponent players from scoring a goal. A player assuming the role of a goal keeper requires very high accuracy with the ball, his speed can be moderate up to 5 units and his kicking capability can be moderate of 8 units. A player in this role does not stray far from the goal post.

Departments: In our simulation of soccer game, we implemented similar departments for each the teams we dubbed Brazil and Italy. Each team is divided into four departments (a) Forward, (b) mid-field, (c) defense, and (d) goal-keeper each having their unique set of duties to be performed. A player in one department cannot perform the duties in another. Each Department has a set of players playing the required roles in that department and we outline them next.

Forward: This department has players assuming the role of forward players. The region of play of this department is in the fore front of the field. The main objective of this department is to score as many goals as possible. This success of this department is calculated by the number of passes they received from mid-field and defense department have been converted into successful goals. Their productivity increase with increase in number of successful passes they receive from the other departments of the same team, number of times the ball has been obtained from the opponents and there is a boost in productivity when any of the received passes have been converted to successful goals. There are normally three players in this department in the normal play of the game. When the game needs to be more aggressive this number can increase to four to increase the number of scoring goals. The extra player is generally taken from the mid-field department. This might make the mid-field department weaker, but the probability of scoring a goal once the ball enters their region increases considerably so there is a trade off by weakening the other department.

Mid-Field: All roles in this department are mid-fielder roles. The region of play for this department is the center of the field. Similar to the duties of a mid-fielder, this department's objective is to support the forward department by passing the ball and supporting the defenders by trying to obtain the ball from the opponents. The productivity of this department is calculated by the number of times they were successfully able to take the pass sent from the defense department combined with the number of times the pass ball were complete to the forward department. They also get a bonus when they obtain the ball from an opponent. Their productivity decreases when the ball is lost to the opponents. This department plays a crucial role on how the other departments in the game perform. There are normally four players in this department. In some cases, when the game norm is attack, it changes. To increase the aggression in the forward department a player is moved from mid-field to forward. If the norm requires the team to be more defensive, a player is moved to the defense department to strengthen it. In both of these cases, there is a trade off in performance for the mid-field. But care has been taken to avoid this decrease in performance affecting the productivity of the department by providing buffers in the calculations.

Defense: The players in this department consist of all players assuming the role of a defender. The region of play of this department is the nearest region to the home goal post. The soccer rules require them to play at a certain distance away from the goal post if the ball is not in the proximity. If the ball is in the proximity of the goal these players are allowed to support the goal keeper in protecting the goal post and preventing advances by opponents. Their objectives are to prevent the opponent from successfully scoring a goal. They have to prevent the opponents from charging into this region with the ball. They should try to capture the ball from the opponents and pass it on to the mid-field region. Their support is essential for the goal keeper to do perform his job well. When the opponents score a goal they are the first ones to receive the ball from the goal keeper. Their productivity is calculated by the number of times they were successfully able to pass the ball from the defense region to the mid-field region and number times they successfully receive a pass from the goal keeper. Their productivity is penalized whenever the ball is lost to the opponents, or they fail to receive a pass from the goal keeper and when the opponents score a successful goal. There are normally three players in this department but when the play is required to be defensive, a player is moved from mid-field to defense to strengthen the defense. This strategy is applied when the lead is comfortable and the team has to focus to win and to defend its current score.

Goal-Keeper: There is only one player in this department. Despite this, we consider it to be a separate department due to the uniqueness of this role. The region of play of the goal keeper is at the home goal post. The prime objective of a goal keeper is to prevent the opponents from scoring a goal. In our implementation, we prevented the goal keeper from moving further into the field and prefer to have him stay at the home goal post. The productivity of the goal keeper is assessed by the number of goals he successfully saved when the ball reaches the proximity of the goal. His productivity is penalized when he fails to pass the ball correctly to his team defenders and when he fails to successfully divert the ball.

Productivity: The productivity of each player is computed using equation 1 cited earlier in this paper. The basis of calculation for productivity is mentioned earlier when we described features of department. The productivity of individual player is calculated and the productivity of the department is calculated based on that. The team productivity is calculated considering the department productivities and the performance of the team as a whole. Productivities of teams are monitored in the graph in the user interface for the simulation.

Utility: The utility of each individual player performing in the designated role is calculated according to equation 2. The individual utility captures how effective each player is in performing in the designated role and the task at hand. We considered a randomly generated synergy while calculating the individual utilities. The department utility is calculated considering how effective those assigned individuals are at performing tasks and promoting the interests of the department. The utility of the team as whole has been calculated by considering the effectiveness of the departments' performance in promoting interests of the team, the synergies generated between the departments of the team, and the overall performance of the team.

Norm: We implemented notions of norm in this simulation. The norm in this game acts as motivations or attitude with which the players play the game. These can be considered to be simulation modes. We designed three norms each differing from others in unique ways.

Norms implemented are Attack, Defend, Self-Gain. In the game, the user (i.e., MOTL controller) prescribes the norm, which governs the game.

Attack Norm: Using this norm, the players are expected to play with attack mode on their mind. The game is modified in such a way that there is 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 a 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 have more capabilities to play in the forward. If it does not implicitly find one, it calls for a reorganization and reassigns the roles to players. The system does not look for the best or worst performance. It just performs reorganization until the user is satisfied with it. Only the user can instruct the simulation to stop reorganization. Therefore, user is the one who controls the extent to which reorganization takes place.

Defend Norm: In this norm, 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 loosing aggression; i.e., 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 continues reorganizing until it receives an instruction from the user.

Selfish Norm: With this norm, players play in the regular department layout configurations. Instead of concern for adaptation, they play to maximize their productivity and to improve their respective utilities. When reorganization is initiated, the system checks if a player is suitable for the role he 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, then he is underperforming so he is replaced. The system maintains reorganizing until user aborts it.

Synergy: In this simulation, we take the average of the department's average productivity and compare it with the individual synergy. If player's productivity is more than the department productivity, we assume he has positive synergy. If he is performing below his capability we assume that he has negative synergy. Each player in the game has a synergy with respective to their own department. Future research is needed to consider in depth integration of synergy to various facets of organization.

5. Observations

Our framework and accompanying simulation are intended to establish a methodological approach for oversight of computational organizations that have become commonplace in complex systems such as in the military. We make no claims about soccer simulation or comparisons to existing high quality simulations. In this section we discuss prototypical usage cases of our soccer simulation along with observations that provide insights pertinent to computational organizations. Our discussion is supported by distinct sets of experiments in three categories we term cases. Since there were many options for norms and modes we opted to group our experiments in such a way to demonstrate only the salient points. Each case constitutes a large class of experiments with similar parametric settings. Our first case establishes baseline observations prior to use of reorganization feature. Subsequent cases establish observations for usage and benefits of reorganization framework.

Case 1
Team Italy settings were: Norm = Defense; Reorganize = Off
Team Brazil settings were: Norm = Defense; Reorganize = Off

In case 1, we allowed simulation to run for a time period of 11400 clock cycles and results were averaged over 1000 runs. Team Italy performed superior to team Brazil.

We compared utilities between the respective teams. Both team norms were set to "Defense". There were four players in the defense department and three players in the mid-field and three in the forward department. These results indicate that over time, productivity of each team was monotonically increasing and as a result of this, utility for each team was increasing. With reorganization switched off for both, there is no reorganization performed. There was very little difference between team utilities. Utility for Italy was 36.9 and utility for Brazil is 35.1. The difference between the utilities of teams is 36.9-35.1 which equals to 1.7 units. Players for each team possess preset parameters at the beginning of game. This simulation showed that the game has been slightly in favor of Italy. Since there is no reorganization in this case there were no reorganization effects to observe. Case 1 is our baseline case and mainly shows that teams are nearly equally matched prior to reorganization.

Case 2
Team Italy settings were: Norm = Attack; Reorganize = On
Team Brazil settings were: Norm = Attack; Reorganize = Off

In case 2, we set the norm for both teams in "attack" mode. Reorganization switch is turned on for Italy and it is turned off for Brazil. There are three players in Defense and Mid-Field department and there are four players playing on the forward department. The simulation was repeated numerous times and on various computer platforms to reach statistical validity.

Averaged over all runs, team Italy's utility was 36.9 and Team Brazil's utility was 31.2. Due to the deployment of the reorganization algorithm, team Italy exhibited a considerable advantage over the opponent. The difference between the utilities for team utilities is 36.9-31.2 = 5.7 units (i.e., nearly 17% advantage). With progression of the game the differences in team utilities monotonically increased as the reorganization algorithm deploys best players for the best possible roles within the bounds of the attack norm until utilities reached a sustained maximum value. Case 2 demonstrated the expected advantages for reorganization.

Case 3
Team Italy settings were: Norm = Self-gain; Reorganize = Off

Team Brazil settings were: Norm = Self-gain; Reorganize = On

In case 3, we changed the norm to Self-Gain or Selfish norm for both teams. The reorganization switch is turned on for Brazil and it is set to off for Italy. In Selfish norm, the game layout stays at default sizes of departments with three players in forward and defense and four players in mid-field. With this norm governing the play, each player attempts to maximize his own utility and productivity even at the cost of overall drop in the team productivity in some cases.

Case 3 sets of experiments were conducted in similar style to previous two cases. The utility of team Brazil was considerably higher at 33.8 and the utility of Team Italy was 27.3 where the utilities differed by about 5 units (nearly 15%). This difference is comparable to case 2. This observation further corroborated the advantage of using the reorganization with a constant norm. We witnessed an overall drop of the team utility at 33.8 as the maximum for the same length of time as the earlier cases. This performance degradation, of about 8 to 9% drop, is due to the choice of selfish norm, which promotes individual utility at some possible detriment to overall utility of the team. This may not be always be the case. When we consider a situation where all the players in the team are exceptionally skilled and are most suitable for their roles, an ideal condition, then the self-interest norm may possibly enhance the overall utility for the team.

6. Conclusions

We have made novel strides in managing oversight of agent organizations that contributes to development of our man on the loop paradigm [22]. In the paradigm that has inspired current research; we endeavor to provide human supervisor capacities to monitor large, complex systems with methods that provide feedbacks that are more natural and intuitive. This enables the supervisor to monitor the system with manageable cognitive overload. Meanwhile, the supervisor is provided with features and parameters that alter system interactions. In present paper, the system is an organization. Our methodology for oversight of organizations allows for a human supervisor to prescribe normative patterns of behavior, which in turn guide the reorganization process. Individuals most fit to play their current roles remain in current roles while others are directed to seek supportive roles to departments that augment their synergy. We have outlined founding principles for computational organizations in our <C, R, D, N> model. We have demonstrated operation of our organizational model using the popular game of simulated soccer and the results are generic and transfer to other domains.

Our future work will include development of rich models of synergy and incorporation to self-organization as well as a varied range of user guided reorganization. This research can be further extended in many directions. Organizations of the *agents* within the agent society can be blended with cultural parameters, which will reveal the full spectrum of power distance index as difference in power among groups of individuals will appear. Moreover the hierarchical distribution of power will allow us to identify key agents which will negotiate with the human supervisor instead of every individual group member. These key agents with higher power can be treated as leaders which can also supervise their subordinate agents. Using this concept in the paradigm we can also develop agent reorganization based on their cultures rather than just their 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 a markov decision process (MDP), we can explore automatic solutions to organizational changes as parametric state transitions in MDP.

Acknowledgement

This research is sponsored by the Air Force Research Laboratory (AFRL) under contract FA8750-06-C-0138. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of AFRL or the United States Government. Special thanks go to the Net Centric SIGINT Focused Information Enterprise (NCSFIE) at AFLR for its support. Author wishes to express gratitude for constructive feedback received from several anonymous reviewers on an earlier draft.

References

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

[2] R. Albert, A. Barabasi, 2002. Statistical Mechanics of Complex Networks. In *Review of Modern Physics*.

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

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

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

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

[7] L. Chaimowicz, M. Campos, R. Kumar, 2002. Dynamic Role Assignment for Cooperative Robots . *IEEE Conference on Robotics and Automation* .

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

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

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

[11] M. Gaston, M. desJardins, 2005. Agent Organized Networks for Dynamic Team Formation. *AAMAS* .

[12] M. Gaston, J. Simmons, M. desJardins, 2004. Adapting Network Structure for Efficient Team Formation. *AAMAS Workshop on Learning and Evolution in Agent Based Systems* .

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

[14] N. Glasser, P. Marignot, 1997. The Reorganization of societies of autonomous agents. *MAAMAW* , 98-111.

[15] H. Handley, A. Levis, 2001. A Model to evaluate the effect of organizational adaption. In Handley, & A. Levis (Eds), Computational and Mathematical Organization Theory, Kluwer.

[16] H. Hexmoor, B. McLaughlan, G. Tuli, 2008. Natural Human Role in Supervising Complex Control Systems, In Journal of Experimental and Theoretical Artificial Intelligence, Taylor and Francis, pp. 59-77.

[17] 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.

[18] P. Kazakos, A. Zaidi, 2008. An Algorithm for activation timed influence nets, IEEE conference Information Reuse and Integration.

[19] 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.

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

[21] E. Matson, S. DeLoach, 2004. An Organizational Model for Designing Adaptive Multiagent Systems. *AAAI Workshop on Agent Organization*.

[22] B. McLaughlan, H. Hexmoor, 2009. Influencing Massive Multi-agent Systems via Viral Trait Spreading, In Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems.

[23] 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.

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

[25] J. Smith, 1996. Influence Net Modeling with causal stregths : An Evolutionary Approach. *Proceedings of command and control research technology symposiun*.

[26] G. Valetto, G. Kaiser, and G. S. Kc, 2001. A moblie agent approach to process-based dynamic adaption of complex software systems. *8th European Workshop on Software process technology, pp*. 102-116.

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

[28] A.K., Zaidi, and P. Papatoni-Kazakas, 2007. Modelling with Influence Networks Using Influence Constants: A New Approach.

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

Author Bio:

Dr. Henry Hexmoor received the M.S. degree from Georgia Tech (1995), Atlanta, and the Ph.D. degree in computer science from the State University of New York, Buffalo, in 1996. He taught at the University of North Dakota before a stint at the University of Arkansas. He is currently an Assistant Professor with the Computer Science Department, Southern Illinois University, Carbondale, IL. He has published widely in artificial intelligence and multiagent systems. His research interests include multiagent systems, artificial intelligence, cognitive science, and mobile robotics.

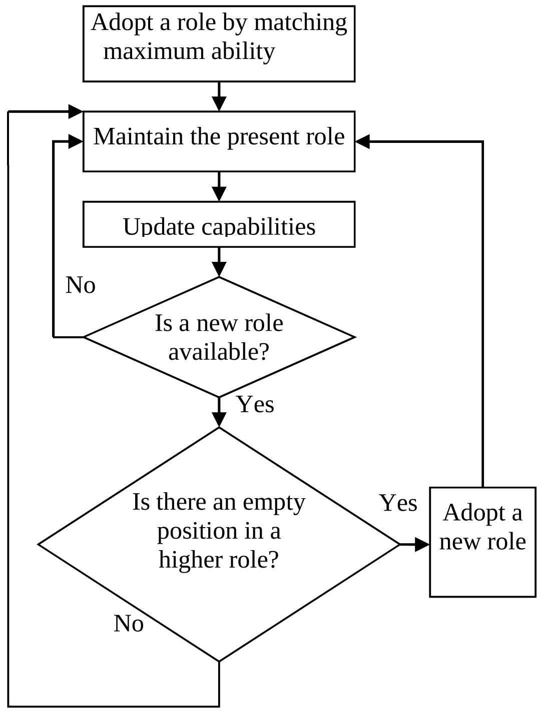

Fig. 1. Flow chart for role promotion using self-initiated norm

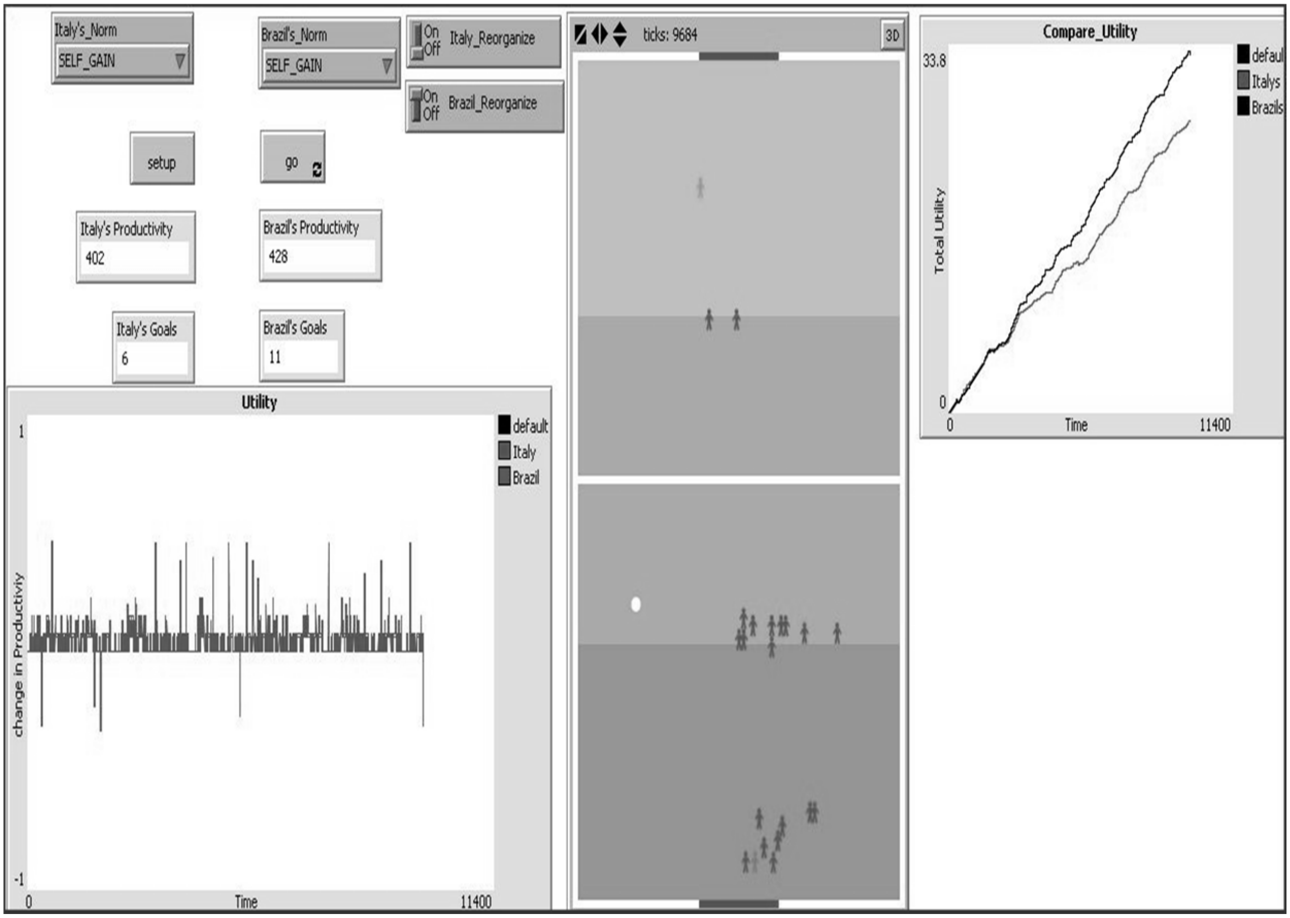

Fig. 2. A typical game screenshot