Strategic Structural Reorganization in Multi-agent Systems Inspired by Social Organization Theory

BY
© 2012
Maryamossadat N. Mahani

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graudate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

____
Dr. Arvin Agah, Chairperson

____
Dr. Prajna Dhar

____
Dr. Swapan Chakrabarti

____
Dr. Prasad Kulkarni

____
Dr. Brian Potetz

Date Defended:___________________

The Dissertation Committee for Maryamossadat N. Mahani
certifies that this is the approved version of the following dissertation:

Strategic Structural Reorganization in Multi-agent Systems Inspired by Social Organization Theory

____
Dr. Arvin Agah, Chairperson

Date Approved:___________________

Abstract

Autonomic systems, capable of adaptive behavior, are envisioned as a solution for maintaining large, complex, real-time computing systems that are situated in dynamic and open environments. These systems are subject to uncertainties in their perceptual, computational, and communication loads. As a result, the individual system components find the need to cooperate with each other to acquire more information and accomplish complex tasks. Critical to the effective performance of these systems, is the effectiveness of communication and coordination methods. In many practical applications of distributed and multi-agent systems, the problem of communication and coordination becomes even more complicated because of the geographic disparity of tasks and/or agents that are performing the tasks. Experience with even small systems has shown that lack of an effective communication and coordination strategy leads the system to no-answer, or sub-optimal answer situations.

To address this problem, many large-scale systems employ an additional layer of structuring, known as organizational structure, which governs assignment of roles to individual agents, existence of relations between the agents , and any authority structures in between. Applying different organizational structures to the same problem will lead to different performance characteristics. As the system and environment conditions change, it becomes important to reorganize to a more effective organization. Due to the costs associated with reorganization, finding a balance in how often or when a reorganization is performed becomes necessary.

In multi-agent systems community, not a lot of attention has been paid to reorganizing a system to a different organizational structure. Most systems reorganize within the same structure, for example reorganizing in a hierarchy by changing the width or depth of the hierarchy. To approach this problem, we looked into adaptation of concepts and theories from social organization theory. In particular, we got insights from Schwaninger's model of Intelligent Human Organizations. We introduced a strategic reorganization model which enables the system to reorganize to a different type of organizational structure at run time. The proposed model employs different levels of organizational control for making organizational change decisions. We study the performance trade-offs and the efficacy of the proposed approach by running experiments using two instances of cooperative distributed problem solving applications. The results indicate that the proposed reorganization model results in performance improvements when task complexity increases.

Acknowledgements

This dissertation would not have been possible without the help, support, and guidance of so many people in my life. I would like to thank all the teachers of my life, from those who thought me the basic aplphabet, to those who taught me how to conduct research and how to solve some complex equationss. From my parents who taught me how to persist and try for the best, to my mentors who gave me their words of advice. From my relatives who always stood behind me, to my friends who gave me hope and support.

I would like to especially thank my graduate advisor, Dr. Arvin Agah, without whom this research and dissertation would not have been possible. Thank you for your continuous support. Thank you for all the challenges, encouragements, suggestions, comments, and critiques which have helped me shape this work and without them I would not have had the opportunity to do this. Thank you for accomodating to my remote work schedule and being willing to spend your weekends working with me. Thank you for your great cooperation and follow-ups. Thanks for being such a positive and happy influence in my academic career.

I would like to thank the members of my committee, Dr. Brian Potetz, Dr. Swapan Chakrabarti, Dr. Prasad Kulkarni, Dr. Prajna Dhar, Dr. Man Kong, and Dr. Sara Campion for all their hard work and participation. Your questions guided me towards interesting and challenging research paths, and our discussions helped me frame the broader context of this research.

I would like to thank all my colleagues who supported me in taking time off work and working on my dissertation. Thanks for beleiving in me.

I would like to thank all the friends who helped me find papers, resources and journals late at night and while away from campus. Alisa Moldovanova, Mark Calnon, Mark Snyder thanks to all of you for your prompt support.

I would like to thank all the staff at the EECS depatment and ITTC, especially Pam Shadoin for arranging all the paper work. Thank you for all the reminders, and thanks for accomodating to my remote studies.

I would like to thank my aunt and uncle, Shirin Nematollahi Saber, and Dr. Al Saber who opened this path for me and always supported me in all aspects. I truely appreciate all your support and your encouragements and I know that I would not have been where I stand today without your unconditional love and support.

I would like to thank my fiance, Dr. Clayton Harper, for his emotional support, encouragements, and for all the effort that he has made to free up time for me to study. Thanks for being a great source of inspiration for me.

My family has been the foundation for my success throughtout my academic career. I would like to thank my mother and father, Nahid and Javad Nematollahi Mahani, for their lifetime support of me and my educational path. Thanks for helping me to strive for my dreams. Thanks for the numerous number of leasons that you have thought me. Without your continuous support and without all those lessons, I would have never been where I am.

And last, not least, thanks to my brothers, Dr. Amir Nematollahi Mahani, and Alireza Nematollahi Mahani, for always being wonderfully supportive and inspiring brothers. Thanks for being there for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The field of multi-agent systems is a common paradigm for design of complex systems. This paradigm is applied to situations which benefit from a collection of autonomous or semi-autonomous entities working in open and dynamic environments. As a result, the agents not only have to cope with uncertainties caused by their own perceptual, computational and communicational limitations, but they also need to take into account any uncertainties imposed by the environment within which they are situated. In such settings, agents will find the need to cooperate with the other agents to acquire more information and cope with the uncertainties, in an attempt to reach a global outgrowth [34]. As the scale of a multi-agent system increases, issues of application performance arise. Critical to the effective performance of these systems are the effectiveness of the communication and coordination.

The interaction in multi-agent systems is governed by some form of explicit or implicit organization which governs the assignment of roles to individual agents, the relationships among the agents, and any authority structures in between. As the scope and scale of multi-agent systems increases, it becomes more important to have a suitable agent organization which can help the system achieve a high performance and reach global goals. Many systems employ an additional layer of structuring, referred to as organizational structure.

It is generally agreed that there is not a single organizational structure that might be appropriate for different applications and domains [8, 23]. Horling and Lesser [24] show that applying various organizational designs to the same problem leads to different performance characteristics. Experience with various multi-agent systems has shown that in each different environment a different decision strategy for agents might work better [62]. For example, in a purely static and unchanging environment, a pro-active behavior might be adequate; while in a dynamic changing environment it would be important to have the flexibility of modifying intentions. At the same time, for performing different task types, a different organizational structure might be more appropriate. For example, while a dynamic coalition formation might be the best organizational structure for detecting the zone of a moving object, a team with a leader might be the best structure for keeping track of a detected object. The reason being that the communication, processing, and memory overload of an initiated task are reduced by keeping the main thread in hands of one leader agent using a team structure, rather than passing the responsibility to others.

As the environment, goals, and system conditions change, it becomes important for a system to reorganize and adapt to the changes. Reorganization can apply to the number of agents, their role and relationships, or agent properties. Regardless of the type of reorganization, there will be costs associated with it. As a result, there needs to be a balance in how often and when a reorganization is performed. One of the issues in reorganization is defining the criteria upon which to evaluate and find a more promising organization to reorganize to. Findings from social organizations emphasize that reorganization efficiency is influenced by the organizational size and complexity [44]. Consequently, it becomes important to gain an understanding of the problem characteristics for which a reorganization can be beneficial.

Different approaches taken by the research community to initiate a reorganization include setting a performance threshold [27], finding inefficient patterns of communication [32], setting a time limit, or a certain number of simulation steps [27]. These approaches can fail to identify all cases for which a reorganization can be beneficial, thus they may not fully utilize the performance gains of a potential reorganization. To date, relatively a small amount of work has been done on reorganizing to a different model of organizational structure. Most systems reorganize within the same structural model [54], [25], [32], [58]. For example, restructuring the agents in a hierarchical model of organization by changing the depth or width of the hierarchy [20, 45].

Theories about human organizations can provide insights to multi-agent organizations. The field of organizational design in multi-agent systems is still in need of powerful tools and methods for enabling effective design, control, and transformation of organizations of different kinds. In this work, we address the problem of reorganization in a distributed problem solving model as a typical multi-agent system application. We look into adaptation of concepts and theories from social organization theory. We propose an organizational model that employs different levels of organizational control for making organizational change decisions. Experimental evaluations are performed using two different applications of pursuit game and cow herding. The experimental results indicate that the proposed model allows the system to stay ahead of the organizational change, resulting in performance improvements.

1.2 Research Hypothesis

We hypothesize that applying a multi-level control mechanism for performing reorganization between different types of organizational structure will reduce the costs associated to reorganization and thus enhance the system performance in cooperative distributed problem solving applications.

1.3 Contributions

Overall, In this work we look into theories and concepts from social organization theory to enhance agent cooperation in geographically dispersed environments. We specifically look into Schwaninger's model of Intelligent Human Organizations [50]. We propose a structural reorganization method which relies on triggering reorganization with changes in task types and employs a multi-level control mechanism for making organizational change decisions in multi-agent systems. The following contributions will be made to the end.

1. We illustrate how applying different types of organizational structure to the same cooperative distributed problem solving application can result in different performance characteristics.

2. We show that it is possible to develop a structural reorganization that allows restructuring between different types of organizational structure and makes possible changes to the organizational entity at run time. This is accomplished by means of creating a two-level control structure that is used for making organizational control decisions and allows interoperation between the two levels, namely the *strategic management* level and the *operative management* level.

3. We demonstrate that the strategic structural reorganization can be more effective than reorganization within the same type of organizational structure when task complexity increases, thus the costs associated with reorganization are balanced by the overall gains of reorganization. This is accomplished by performing experiments which include both methods.

We will return to this list of contributions to provide additional details about how they have been accomplished.

1.4 Dissertation Outline

This dissertation is organized into seven chapters. Following Chapter 1 on Introduction, Chapter 2 provides an overview of social organization theory and some of its concepts.

In Chapter 3, copperative distributed problem solving is introduced and the issues of coordination and cooperation in multi-agent systems are discussed. Overview and discussions of the related work on application of organizational structures and reorganization in multi-agent systems are also provided in this chapter.

In chapter 4, the research methodology and approach to the problem of reorganization in multi-agent systems are further discussed. The research hypothesis and the solution characteristics are also elaborated on.

In chapter 5, the details of applying our proposed solution to a pursuit game as a sample cooperative distributed problem solving application are provided. The organizational modeling language, organization and reorganization models, details of the simulation and experimental setup are provided. Results of the experiments are presented and discussed at the end.

In chapter 6, details of applying our proposed solution to a cow herding scenario taken from the 2010 multi-agent programming contest as another sample cooperative distributed problem solving application are provided. The organizational modeling language, organization and reorganization models, details of the simulation and experimental setup are presented. Results of the experiments are presented and discussed at the end.

Chapter 7 provides a summary of the work. This chapter also describes the contributions, lessons learned, limitations, and future work.

1.5 Glossary

The terms used throughout this dissertation are defined in this section.

- **ACL** - Agent Communication Language.
- **Agent** - An autonomous object with the ability to perceive, reason, and act. An agent has the ability to communicate with the other agents in a system through a common communication language.
- **AgentSpeak** - An agent-oriented programming language. It is based on logic programming and the BDI architecture for autonomous agents.
- **BDI Architecture** - The BeliefDesireIntention architecture is a software model developed for programming intelligent agents. Superficially characterized by the implementation of an agent's beliefs, desires and intentions, it actually uses these concepts to solve a particular problem in agent programming.
- **Coalition** - A set of agents that work together to solve a joint problem.
- **Coalition Formation** - The process of coordinating actions of agents to form a coalition for solving a joint problem.
- **Cooperative Distributed Problem Solving** - A network of semi-autonomous processing nodes working together to solve a problem, typically in a multi-agent system. That is concerned with the investigation of problem subdivision, sub-problem distribution, results synthesis, optimization of problem solver coherence and co-ordination. It is closely related to distributed constraint programming and distributed constraint optimization.
- **Distributed Sensor Network** - A collection of a large number of heterogenous intelligent sensors which are distributed logically, spatially, or geographically over an environment and connected through a high-speed network. In this work the terms sensor network, sensorNet, and Distributed Sensor Network (DSN) are used interchangeably.

- **Doppler Radar** - A radar which takes use of doppler effect to create data about objects at a distance. A doppler radar beams a microwave signal towards a desired target and analysis its reflection to perform measurements of a target's velocity relative to the radar source and the direction of the microwave beam.

- **Dynamic Environment** - An environment in which events and phenomena occur and change the environment. In the context of a sensor network, a dynamic environment could be caused by changes in resources such as bandwidth, number of agents, etc. In this work, a dynamic environment for the sensor network simulations refers to unpredictable appearance and disappearance of sensor nodes.

- **Heuristic** - An algorithm that is able to produce an acceptable solution to a problem in many practical scenarios, in the fashion of a general heuristic, but for which there is no formal proof of its correctness. Alternatively, it may be correct, but may not be proven to produce an optimal solution, or to use reasonable resources. Heuristics are typically used when there is no known method to find an optimal solution, under the given constraints (e.g., time, space) or at all. Often specially crafted problem instances can be found where the heuristic will in fact produce very bad results or run very slowly; however, such pathological instances might never occur in practice because of their special structure. Therefore, the use of heuristics is very common in real world implementations. For many practical problems, a heuristic algorithm may be the only way to obtain good solutions in a reasonable amount of time.

- **Holon** - A self-similar or fractal structure that is stable and coherent, and has integrity and identity at the same time as it is a part of a larger system, it is a subsystem of the larger system.

- **Jason** - An Open Source interpreter for an extended version of AgentSpeak. It implements the operational semantics of that language, and provides a platform for the development of multi-agent systems. It is distributed under GNU LGPL.

- **J-MOISE+** - J-MOISE+ is an open source organizational middleware that follows the MOISE+ specification and uses Jason agents.

- **Macro-structure** - The overall organization of society, described at a rather large-scale level, featuring for instance social groups, organizations, institutions, nation-states and their respective properties and relations. In this work, it refers to the complete system or organization.

- **Micro-structure** - A structure on a small scale. In this work, it refers to an individual agent or a sub-group of agents as compared to the whole structure.

- **MOISE+** - Model of Organization for multIagent SystEms is a framework which provides a rather complete infrastructure for modeling organizations.

- **Moving Target Indicator** - A radar with a Moving Target Indicator (MTI) functionality is able to distinguish between real stationary objects, real moving objects, and ”clutter”, or electronic noise giving a false impression of a target.

- **Multi-agent System** - A system composed of multiple interacting agents.

- **NEXP-Complete Problem** - A Nondeterministic Exponential Problem (NEXP) is a highly intractable problem which can be solved nondeterministically in exponential time.

- **Open Environments** - The most complex general class of environments which are continuous, non-deterministic, dynamic, and inaccessible.

- **Organization** - A system composed of interacting agents that have relationships. An organization works towards some high-level goals, such as supporting numerous operations, or increasing productivity.

- **Organizational Cybernetics** - A systems approach which applies the principles associated with communication and Control from Cybernetics, to organisations. Organizational Cybernetics has been developed from both a theoretical and methodological point of view.

- **Organizational Structure** - The architecture of a multi-agent system which covers the pattern of information and control relationships between agents. The organizational structure specifies assignment of roles and responsibilities to agents in a problem-solving or cooperative planning effort.

- **Organizational Performance** - The actual output or results of an organization as opposed to its goals.

- **Platform** - Software or application framework.

- **Real-time Environment** - An environment in which correctness of a result is highly dependent on the time at which the result was produced or an action was taken. For a system to handle requirements of a real-time environment, it is not enough to take actions or produce results quickly, but these should be done at the right time. In the context of sensor networks, as these networks operate in real-world, they will have explicit real-time constraints related to the environment.

- **Reorganization** - Changes to the organizational structure so that it becomes more suitable for a new goal or task type.

- **SACI** - Simple Agent Communication Infrastructure is a Java API and a set of tools that can be used in order to help the development of societies of distributed agents.

- **S-MOISE+** - SACI - Model of Organization for MultIagent SystEms is an open source implementation of an organizational middleware that follows the MOISE+ specification. It is an extension of SACI where the agents have an organizational aware architecture.

- **Strategy** - A plan of action designed to achieve a particular goal. In this work, it refers to a plan of action that determines what organizational structure will be used for each task type. A strategy will let the system decide how it will allocate resources and roles to the agents, and how the agents will be interacting and cooperating for a specific task type and based on conditions of the environment and characteristics of the current task.

Chapter 2

Social Organization Theory

Human organizations often perform in ways that can be considered intelligent. They can adapt to changes and uncertain environments, they can learn from their own and others experiences, they can diagnose existing problems or foresee upcoming problems and take action to manage them. To some extent, this is achieved by intelligent individuals, but a certain amount of the intelligence is contributed by an organization that they adhere to and its structure.

Herbert Simon [53] refers to an organization as the "pattern of communications and relations among a group of human beings, including the process for making and implementing decisions." The main importance of these patterns is that they feed the organization members with the information, assumptions, goals, and attitudes that will affect their decisions. These patterns also provide the members with a set of comprehensible and stable expectations on the actions of other group members and how they should react to them.

An organization is composed of organizational entities whose coordination is implemented through information flows and exchanges. These organizational entities include:

- Tasks, Parts, Tools, ... (the answers to *what*)
- Functions, Processes, ... (the answers to *how*)
- Schedules, Schedulers, ... (the answers to *when*)

- Locations, Destinations, ... (the answers to *where*)

Social organization theory research has addressed various aspects of organizational structures and reorganization. There are two main areas of research in social organization theory that we are interested in: *organizational process model*, and *organization development* [2].

- The organizational process model is about processes that organizations use when encountering situations that will require change. Allison [2] introduces te following propositions for organizational process model in social organizations:
  - ”When faced with a crisis, leaders do not look at it as a whole, but break it down and assign it according to pre-established organizational lines.“
  - ”Because of time and resource limitations, rather than evaluating all possible courses of action to see which one is most likely to work, settle on the first proposal that adequately addresses the issue, which Simon [53] termed ”satisficing”.“
  - ”Gravitation towards solutions that limit short-term uncertainty (emphasis on ”short-term”).“
  - ”Organizations follow set ”repertoires” and procedures when taking actions.“
  - ”Because of the large resources and time required to fully plan and mobilize actions within a large organization, effectively stay limited to pre-existing plans.“
- ”Organization development is about creating a response to change, a complex educational strategy intended to change the beliefs, attitudes, values, and structure of an organization so that it can better adapt to new technologies, markets, challenges, and the dizzying rate of change itself.”

2.1 Social Organizations and Organizational Structures

Social organizations can be categorized based on many different elements. Span of control, and division of skills and expertise are one group of components based on which organizations can be classified into different types. These organizational types are:

- Functional structure, in which people with similar skills are grouped together and managed by someone who knows about those skill sets.
- Divisional structure, in which groups of people with similar skills are spread across the organization where they are needed.
- Matrix structure, that is a combination of functional and divisional structures in which teams of people are used to take advantage of strengths and reduce effects of weaknesses of functional and divisional structures.
- Horizontally linked structure, in which people are grouped along activities and processes.

Each of these structures has its own advantages and disadvantages. For example, in an organization with a functional structure siloing can become a problem when various departments become isolated from each other and do not communicate. This disadvantage makes the functional structure more appropriate when there is not much need for intra-divisional communications. Some examples of structural problems in organization theory and solutions for them are:

- **Functional structure** The structural problem is how to increase and facilitate sharing expertise for a particular functional activity. The solution includes building teams that are based on common-functions in a bottom-up manner, and managing those teams by leaders who have in-depth knowledge of the function. The result is a set of functional units that are controlled and coordinated from the top management.

- **Mechanistic structure** The structural problem is how to maintain strict control of the organization in order to ensure efficiency. The solutions for this problem include tendency toward functional structure with high span of control, extensive division of labor, high degree of formalization, and conflict resolved through hierarchical channels.

- **Matrix structure** The structural problem is how to differentiate the organization's structure and processes to contain the stable and dynamic areas of operation. The solutions for this problem include use of (1) loose matrix structures which can combine both functions and product groups; (2) moderately centralized control system with access to feedback loops that are both horizontal and vertical ; (3) performing conflict resolution through product managers or by means of normal hierarchical channels.

- **Organic Structure** The structural problem is about facilitatating and coordinating numerous and diverse operations. The solutions include tendency toward product structure low span of control, low division of labor and low degree of formalization; decentralized control and conflict resolved through integrators.

Several structures from social organization theory have been successfully applied to multi-agent systems. A brief summary of each of these structures follows:

- **Hierarchy** Supports a tree-like structure in which every entity in the organization is subordinate to another entity. A hierarchy includes an individual/group with power at the top. Subsequent levels of power follow beneath them. This is a dominant mode of organization among other organizational structures. This structure can be similar to divide and conquer approaches, breaking down a problem into sub-problems.

- **Holarchy** This structural model can be considered as nested hierarchies of self-replicating structures. Holons can form several levels of resolution in a holarchy and perform as autonomous wholes and yet cooperate as a whole to achievie the goal of the holarchy. Within a holarchy, holons can belong to different clusters simultaneously. The holons follow a rule-governed behavior. The rules define a system as a holon with an individuality of its own; they determine invariant properties of a holon, the structural configuration of the holon and its functional pattern.

- **Coalition** A treaty among individual entities or groups, during which they cooperate in a joint action, and each with their own self-interests, for a common cause. This structure might be temporary. A coalition can also be considered as a means-oriented arrangement which allows distinct entities to pool resources together and combine efforts to effect change. The form of coalition, its type and duration can be distinctive factors between them. Some examples are:

  – Campaign coalitions will have high intensity and long duration.

  – Federations will have lower degree of involvement and intensity, but still with a long duration.

  – Event-based coalitions will have a high level of involvement and potential for future cooperation.

- **Team** A team gets formed from a set of cooperative entities with a common goal. The quality of the overall organization is dependent on competency of the constituent teams. A team can have a leader which directs the actions of other team members and performs task assignments to ensure that the team is working towards the common goal. This structure can have a certain disadvantage that is, the team leads can be partial to their own team's needs and result in conflict among the teams. A team structure scales well so it can be a good structure for larger scale cooperation problems.

- **Congregations** Groups of entities who bound together in a typically flat organization in order to derive additional benefits. The group does not seek a single specific goal and it can be a long-lived group. Other standard allocations such as coalitions or auctions can be used within a congregation to decide which agents should perform patricular tasks.

- **Society** A group of related entities which are bound together through persistent relations or a large social grouping, sharing the same geographicall or virtual territoty. All members of the society will be subject to the same dominant cultural expectations. A society is a collaborative entity which enables its members to benefit in ways that would not be possible on an individual basis. Members of a society can have different goals or different levels of rationality. There are guidelines by which the members must act. The guidelines will cause a certain level of consistency and facilitate coexistence of members.

- **Market** This structure accomodates to self-interested entities with individual goals. The interaction between market enties is based on negotiation and communication. A particular agent in the market coordinates activities of the group. Markets are composed of two main roles, buyers which bid for a common set of items, and sellers which process the bids and determine the winner.

2.2 Organizational Flexibility

Vital to the successful operations of an organization is organizational flexibility. Organizations get encouraged to change their structure in order to be able to respond to *trigger points* defined as ”an external event that has an impact on an organization” [18]. A single organization might find it more beneficial to use different structures within different parts of the organization. For example, while a research and development division might benefit most from a matrix structure, a marketing department might benefit most from a functional structure. The difference in how organizations perform division of resources, makes them appropriate for different tasks and different environmental conditions. For example, a matrix structure is most efficient when resources are scarce because it makes sure that the scare expertise and skills are being used full time and for the most critical tasks. A divisional structure might cause redundancy of efforts and resources because of the parallel activities that could happen between different divisions. Thus for efficient selection of an appropriate organizational structure, it becomes critical to have a good understanding of different organizational structures and what makes them appropriate for certain tasks or circumstances.

Organization development (OD) is a deliberately planned effort to increase an organization's relevance and viability. OD can be referred to as future readiness to meet change, thus a systemic learning and development strategy intended to change the basics of beliefs, attitudes and relevance of values, and structure of the current organization to better absorb changes. In other words, OD is the framework for a change process designed to lead to desirable positive impact to all stakeholders and the environment. OD can design interventions with application of several multidisciplinary methods and research besides traditional OD approaches.

2.3 Intelligent Human Organizations Framework

Schwaninger's framework of Intelligent Human Organizations [50] conceives the management of organizations as a recursive, multi-level process in which the organization components are dynamically interrelated. Using this framework, an integrated view of the organization is attained by combining three theories from organizational cybernetics:

- The Model of Systemic Control which provides a framework for a comprehensive control of the activities of an organization to enhance its fitness. This activities dimension creates an ensemble of intended organization operations and is formed of the goals, principles, and rules that govern the internal and external behavior of the organization.

- The Viable System Model which addresses issues of diagnosing and designing the structures of an organization for viability and development. This behavioral dimension controls the qualitative features of an organization which govern properties such as reframing or revitalizing the organization.

- The Team Syntegrity model which furnishes a structural framework for developing interactive behavior in an organization to enable cohesion, synergy and knowledge creation. This structural dimension handles the mutual interrelationships between the organization components.

In the next chapter, we will review the problem of coordination in multi-agent systems and propose how insights from the stated theories in social organizations will be used to develop a solution.

2.4 Relation between Structure and Strategy

Many theorists believe that strategy and structure are related [43], and the long-term performance of an organization hinges on their relationship. In this context, strategy refers to a plan of action designed to achieve a particular goal. At the same time, strategic behavior should be based on a dynamic model for strategy formation which takes situational factors into consideration, such as crisis/no crisis in performance, presence/absence of a strategic micro-structure, power dependency of inter-organizational relations, goals and resources of the system, etc. The strategic behavior affects the operations on a macro-structure, which determines the performance of the organization. In the current multi-agent systems, such a relationship is completely missing.

Chapter 3

Cooperative Distributed Problem Solving

Cooperative Distributed Problem Solving refers to a loosely coupled network of problem solvers which are working together to solve problems that may be beyond the capabilities of individual agents. Individual nodes might not have sufficient resources, expertise, or information to solve the problem individually, but together they can form groups and cooperate to accomplish tasks and gain a higher performance. Multi-agent systems has been a common platform for investigating distributed cooperative problem solving.

This chapter provides a background on distributed problem solving, the issue of coordination and cooperation in multi-agent systems, and application of organizational structures and reorganization in multi-agent systems. The first section presents an overview of cooperative distributed problem solving area, coordination and cooperation in multi-agent systems and its challenges. Then we provide an overview of organizations and reorganization in multi-agent systems with discussssions of the challenges and the related work.

3.1 Coordination in Multi-Agent Systems

Coordination is an integral part of a multi-agent system. In a multi-agent system, interdependencies among agent activities might rise from the need to have shared resources or to put efforts together to solve a larger problem. Division of a problem to sub-problems, dealing with overlapping sub-problems, or even impossibility of dividing a problem to appropriate sub-problems [36, 35], require having access to efficient methods to select which agent, how, and when to assign tasks to. Durfee [13] defines agent coordination as "an agent's fundamental capability to decide on its own actions in the context of activities of other agents around it". With coordination, the agents can intentionally combine their efforts and resources together and try to accomplish global goals.

3.1.1 Importance of Agent Coordination

Existence of inter-dependencies among agents and the choice of coordination method affect the overall performance of a system [34]. Experience with even small multi-agent systems has shown that lack of an appropriate coordination strategy might lead the system to no-answer or suboptimal answer situations. One example of this problem is shown in works related to Hearsay system. The experiments with this system have shown that not having a general view of the activities of the other agents which are involved in an interrelated subproblem can lead to some degree of incoherence among agents [33]. Rederiving results that were already achieved by other agents or getting distracted by another agent's unreliable results that was based on partial solutions to local problems are samples of such coherence issues.

3.1.2 Coordination Challenges

Coordination between the agents becomes rather complicated when dealing with systems that are inherently heterogeneous and have rather strict time scales. At the same time, in a multi-agent system, the agents' behaviors and actions are affected not only by their own internal properties but also by the properties of their environment. The coordination challenges of multi-agent systems can be investigated in terms of three main properties of *Agent Population*, *Task Environment*, and *Solution Characteristics*. A brief overview of theses properties follows [13]:

- **Agent Population** - Properties of the population of agents that will be involved in cooperation is one of the influential points about a coordination strategy. The main challenges in this regard are:

  – Quantity - Scalability of the coordination strategy with regard to the number of agents. The number of possible interactions between the agents will grow exponentially if each agent is supposed to interact with all the other agents. Bandwidth and computational limitations limit the coordination search problem.

  – Heterogeneity - In a population of heterogeneous agents, the difference in communication language, ontology, and internal architecture of agents should be considered. It is important to consider how well a coordination strategy scales in regard to increasingly heterogeneous populations.

  – Complexity - Complexity of the population of agents that will be involved in a coordination process. Whether the agents are specialized in a specific task, or they are flexible to decide for themselves what goals to reach or how to reach the goals strongly affects the coordination strategy. Coordination between specialized agents can be easier.

- **Task Environment** - Different characteristics of the environment, and the nature and requirements of the tasks influence the coordination strategy. For example, real time tasks might introduce some complications in the coordination strategy.

  – Degree of interaction - Several agents might be interacting to settle an issue (task or sub-task). The number of issues that one agent becomes involved in can increase. Committing to one issue can affect how the agent will settle other issues. This can cause dependencies. The web of dependencies can grow and subsequently the coordination strategies will have difficulty scaling.

- Dynamics - Dynamic nature of an environment might lead to changes in agents' goals or the way of achieving previously set goals. Coping with environmental changes, changes in information, tasks, availability of resources, number of agents and their capabilities is another challenge in agent coordination.
- Distributivity - This point refers to the environments with distributed agents in which origination of the tasks is also distributed. Such distributivity complicates the coordination strategy as it brings uncertainty about which agent is doing what task.

- **Solution Characteristics** - Various characteristics of the desired solution can influence the choice of coordination strategy.
  - Quality - The required quality of the solution in terms of timeliness, efficient use of resources, and efficiency of the coordination.
  - Robustness - How robust the strategy is in dealing with environmental changes, or how well it can deal with deviations from its expectations.
  - Overhead limitations - Some environments might have computation requirements, communication overhead, time limits, etc. How well a coordination strategy can adapt to environments that impose stringent limits is important.

3.1.3 Geographic Disaprity in Cooperative Distributed Problem Solving

In many practical applications of multi-agent systems, agent coordination becomes more complicated because of the geographic disparity of the tasks that need to be performed and their dependencies. One example scenario is a team of agents trying to find and guide a set of geographically dispersed moving targets into a specific location. An exploration task is required to identify and locate the target(s). A directing task is required to guide the targets into the desired destination. These tasks entail dependencies; a guidance cannot be done until a target is identified and located.

There will likely be more than one group of agents performing any of the tasks at a time, and each group might benefit from a different structure of agents. These necessitate coordination on a group level. Since the targets are distributed geographically and only have access to local information, the coordination and cooperation methods need to take into account locality to minimize the time and resources spent on traveling.

3.1.4 Operational Versus Organizational Control

Coordination among agents can be performed using two different perspectives. In an operational control model, which is an agent-centric view of coordination, the decision making is based on the short-term view of agents. As a result, these models rely on a limited and dynamic perspective of the system. Organizational control is based on a long-term view of the system. An organizational control model is based on a global perspective of system performance and is maintained and acheived by means of an organizational structure. Most of the research in agent coordination is based on operational control rather than organizational control [54].

Organizational control and use of an explicit organizational structure positively affects achievement of organizational objectives as these goals can be in a wider degree than what each individual agent can perceive [10]. An organizational structure has knowledge, culture, memories, history, and capabilities that are distinct from the ones for each individual agent. An organizational structure does two major actions:

- Defines roles, responsibilities, and preferences.
- Identifies control and communication patterns.
  - Who does what for whom? Where to send which task announcements and allocations?
  - Who needs to know what? Where to send which partial or complete results?

The ability to develop effective multi-agent organizations is key to development of larger, more diverse multi-agent systems. In this work, we focus on employing an organizational control model in order to keep a long-term view of the system and to be able to consider an overall performance.

Next section brings up the background and related work in regards to applications of organizations in multi-agent systems.

3.2 Organizations in Multi-agent Systems

Interaction in multi-agent systems is governed by some form of implicit or explicit organization. These interactions usually govern some of the system's behaviors, such as authority structures, data flows, and requirement relations. Horling and Lesser [23] define the organization of a multi-agent system as "the collection of roles, relationships, and authority structures which govern its behavior". The organizational design takes control of choosing and moving between different sets of agents, architectural forms, and resources for tasks. Having an organizational structure can be helpful to limit the scope of interactions between the agents, handle uncertainties in the system, reduce or increase a system's redundancy, and form global goals that individual agents may be unaware of [35]. At the same time, an organizational structure can be limiting by reducing the overall flexibility of the system, adversely affecting the computation or communication overhead, and adding one level of complexity to the system [23, 25].

The short and long-term performance of a system are affected by its organization [8, 49, 14, 25, 42, 57, 6]. Applying different organizational designs to the same problem will lead to different performance characteristics [24]. Any method that can be used to compare different organizational designs, understand their behaviors, and reorganize the system to the appropriate structure, will be beneficial.

There are several organizational paradigms that are suitable for and have been applied to multi-agent systems. Hierarchies, Holarchies, Coalitions, Teams, Congregations, Societies, Federations, Markets, and Compound organizations are some examples. Each of these structures has its own specifications, advantages and limitations, such as supporting individual versus group rationality, myopic behavior (an agent not considering critical needs of the other agents, when other resources can still be satisfying for the agent), long-term or short-term structure, and being suitable for homogeneous or heterogeneous populations of agents. It is generally agreed that there is not a single organizational structure which might be appropriate for different applications and domains [8, 23]. It might be even necessary to apply a combination of various structures within one system. These organizational structures, their advantages and disadvantages are described in a survey by Horling and Lesser [22]. Table 3.1 presents a number of these organizational structures that have been applied to multi-agent systems, and lists the main specification of each structural model together with its key advantages and disadvantages.

3.2.1 Organization Formation

Definition of an organizational structure can be implicit and embedded in the design of individual agents, or it can be explicit and defined at the system level. The choice for one or the other depends on characteristics of the application domain [12]. An organization can be formed in different ways. In organization theory, an organization might either emerge spontaneously through interactions of a collection of individual decision makers, or it can be as an already existing and predefined structure into which the individuals should try to fit themselves. In multi-agent systems these two trends exist as well. In some cases, an organization's structure is represented as a series of rules. These rules cover the answer to what, whom, and how to structure the communication. In other cases the focus is on emergent formation of an organization, in which it is tried to have the organizational structure embedded into the agents. A broad classification of the available techniques of organization formation in multi-agent systems follows [23]:

- *Scripted* - An organization is formed based on some statistical predefined instructions.

Structure	**Characteristic**	**Advantage**	**Disadvantage**
Coalition	Dynamic, goal directed	Exploit strength in numbers	Short term benefits may not outweigh organization construction costs
Congregation	Long-lived, utility-directed	Facilitates agent discovery	Sets may be overly restrictive
Federation	Middle-agents	Matchmaking, brokering, translation services, facilitates dynamic agent pool	Intermediaries, become bottlenecks
Hierarchy	Decomposition	Maps to many common domains, handles scale well	Potentially brittle, can lead to bottlenecks or delays
Holarchy	Decomposition with autonomy	Exploit autonomy of functional units	Must organize holons, lack of predictable performance
Market	Competition through pricing	Good at allocation, increased utility through centralization	Potential for collusion, malicious behavior, allocation decision complexity
Society	Open system	Public services, well defined conventions	Potentially complex, agents may require additional society-related capabilities
Team	Group level cohesion	Address larger grain problems	Task centric
Compound	Concurrent organizations	Exploit benefits of several organizational styles	Increased sophistication, drawbacks of several organizational styles

Table 3.1: Various organizational strutures, their main specifications, advantages, and disadvantages.

- *Controlled* - An individual or a group of individuals explicitly apply an organization to a population.

- *Emergent* - There are no central or global directions. The organization emerges through individual actions of agents. These methods are self-directed or organically grown.

In practical applications it might not always be possible to specify one class for a system. In many cases, a combinatory mixture of the above methods might be used. The emergent structures seem to be getting more attention from researchers in modeling complex systems that are embedded in uncertain and dynamic environments. The emergent or bottom-up approaches to self-organization have the disadvantage of being prone to having lower quality than a carefully designed organization. At the same time, emergent organizations are prone to not unfolding because of time constraints.

3.2.2 Organizational Modeling Frameworks

Applying organizational constraints explicitly into a multi-agent system can be acheived by means of an underlying organizational modeling framework. There are several existing frameworks in the literature for modeling organizations. Some examples of these frameworks are: AGR (Agent, Group, Role) [17], MOISE+ (Model of Organization for multI-agent SystEms) [26], ISLANDER [16], Organizational Design Modeling Language (ODML) [23], KB-ORG [54], Virtual Design Team (VDT) [37], and OMNI (Organizational Model for Normative Institutions) [11]. These frameworks differ in various aspects such as internal or external representation of the organization, existence of a separate or distributed management, coverage of structural, functional, and normative aspects of the organization [30]. Further details and description of each of these frameworks follows:

AGR: In AGR, the organization is presented by means of a meta-model which defines a structural relationship between a collection of agents. This structure is defined by means of Agents, Groups, and Roles (AGR). In this model, an agent can be part of more than one group. An agent will have specific role within each group. These groups are created by agents, and the agent who creates a group assumes the role of group manager. Each group is structured by means of two major components: a tuple which contains all the possible roles, and an interaction graph which specifies the interactions between the roles in the group. A main characteristic of this model is its minimalist structure-based view of agent organizations. The set of groups and the possible interactions between the roles that belong to different groups, form the organization's structure. This model provides a reasonable presentation for an organization with several groups in which the organizational structure determines the interaction between the members.

ISLANDER Esteva *et al.* [16] have introduced IISLANDER, a declarative language for specifying electronic institutions, which are equivalent to human institutions in concept with computational applications. Institutions determine how interactions of a certain sort will be structured in an organization. An electronic institution in ISLANDER is formed by means of four basic elements: dialogic framework, scenes, performative structure, and norms. The roles that agents can take and their relationships are defined by means of the dialogic framework. The roles define patterns of behavior within the institution. A collection of agents playing different roles in interaction with each other forms a scene. Every scene specifies the set of possible dialogic interactions between roles instead of agents. The performative structure establishes relationships between the scenes. The commitments, rights, and obligations of agents are specified by means of norms. Since all these items are defined during design time and cannot change during run time, this model does not have enough flexibility for reorganization modeling.

KB-ORG Sims *et al.* [54] have employed a knowledge based approach to the problem of searching for the best organization design. KB-ORG prunes the search space of the best organizational design by use of a knowledge base. This model is very complex and requires a detailed specification of the organization's requirements. KB-ORG not only relies on coordination level organization design knowledge, but also application level organization design knowledge. This model provides a limited variety of organizational structures. The organization structure in KB-ORG is expanded by adding various levels of hierarchy or peer-to-peer relations to the system, if needed. It does not choose between various organizational models. KB-ORG starts a new coordination goal when a task is split between several agents. It is important to be able to apply coordination and reorganization even without the event of a new task split between agents. KB-ORG is focused on assigning agents to roles, while in this work we focus on reorganizing by changing roles and enabling agents to enact roles as the task conditions change. KB-ORG is also not able to deal with any time-varying organizational requirements or environmental expectations.

MOISE+ In MOISE+ (Model of Organization for multIagent SystEms), an organization is formed based on three main aspects of structural, functional, and deontic. The agents' relations are defined by the structural aspect and based on the concepts of roles, groups, and links. The functional aspect determines how the organization achieves its global goals, this includes goal decomposition (plans) and task distribution (missions). A social scheme is used to specify global goals, plans, and missions. The permissions for various roles, and the obligations for various missions are specified by the deontic aspect. This model provides a rather complete infrastructure for modeling organizations. In a more higher level view, the organizational structure in MOISE+ is modeled in form of a graph that is defined as a set of roles, links, and groups. Each role is composed of a set of missions. A mission is a permitted behavior in the system, defined by a set of goals, actions, plans, and resources. Assignment of a role to an agent causes the agent to follow the permitted behaviors specified by the missions of that role. The interaction between the roles is specified by means of the organization links. Three types of organization links are defined: communication, authority, and acquaintance. A communication link specifies the kind of communication that can exist between the roles, the protocols that have to be followed, and any particular missions for which they can be used. The subordination of roles is determined by use of the authority links. The authority links also specify the context within which the subordinate relation is valid. The context is defined by means of the missions that are associated with the link. All the roles about which an agent can possess information and can use in its decision making mechanism, are specified by acquaintance links of a role. A group is composed of a set of roles, missions, and the links that exist between the roles which belong to the group. The MOISE+ model provides a good insight into the influence of a structure on the organization's performance by means of the ideas of relations and interactions with their corresponding graph. When missions are assigned to roles by means of deontic links, it is an implicit interaction protocol. The AGR model, compares only to the structural aspect of MOISE+. Considering that the structural aspect of MOISE+ extends AGR, we will use the ideas from MOISE+ to model our organizational design in this work.

S-MOISE+ - This framework [28] is based on and is very similar to the MOISE+ framework. However, the base idea is an agent that always does what its organisation needs, it does not have personal goals. This model is targeted towards reactive agents.

J-MOISE+ - As discussed, the organization of a multi-agent system can be either embedded in the design of agents, or an explicit organization is defined for the system. The J-MOISE+ framework combines these two models by supporting both an explicit representation of the organization available to the agents at runtime, and an implicit organization by enabling the agents to read, represent, and reason about the organization. This feature makes the J-MOISE+ framework more complete in terms of satisfying the requirements of a complete organization, compared to the other organizational design frameworks. In the other frameworks of agent organizations, only one type of organizational implementation is supported. In this work, we need to employ a model which supports both an explicit and implicit organization so that any knowledge transfer could take place between the two levels of organization. In this work, we will be employing the J-MOISE+ framework for the implementation purposes.

NMAS - Vazquez and Lopez [61] [11] have developed a framework that supports agent reorganization. Their framework has a norm based approach for design of hierarchical organizations. In the NMAS model, each role should have a position profile with it. An agent can change its norms to conform to a specific positional profile. This is how reorganization happens in this model, by change of roles at run-time. This model can be very useful for open systems that have external agents. Still, the model requires all the positions and specifications of the role to be specified at the outset itself. In this work, we seek a more general model of reorganization, rather than just the change of agent roles in a hierarchical structure.

ODML The Organizational Design Modeling Language [24] is a mathematical modeling language which enables modeling various organizational structures such as federations, coalitions, hierarchies, etc. The organizational models produced using this mathematical model can be quantitatively compared against each other. The drawback of this model is that using ODML it is difficult to develop efficient techniques for searching various organizational spaces that can be encountered efficiently. At the same time, ODML requires a significant amount of domain knowledge and effort to build the models; however, as noted by the author, this is not always possible to do, because the predictive techniques have not been discovered or those that are known are insufficiently accurate.

VDT The Virtual Design Framework is designed with the goal of developing a computational model of real life project organizations. The organizational model in VDT is composed of two structures, a communication and a control structure. The communication structure specifies who can talk to whom, while the control structure determines the authority relationships, supervisions, and reporting responsibilities. The agent duties in this model are fixed. The model is also limited by the fact that it only supports hierarchical structures.

We consider an organization as an already existing and predefined structure that the agents need to conform to. As discussed above, we found MOISE+ more suitable for designing our problem solving agent organization. This framework enables us to make sure that the agents within the system follow the organizational constraints.

3.2.3 Organization Selection

In a multi-agent system with a constant number of agents (n), if the agents are allowed to take on multiple roles, then a candidate organizational structure which would contain all these agents will have n^n possible assignments of agents to roles. Even in a single role structure, there would be $n!$ possibilities assuming distinguishable agents. The problem of finding the optimal structure becomes untraceable. The methods to deal with the problem of searching for the most effective organizational structure can be grouped into algorithmic and heuristic solutions. The algorithmic solutions guarantee finding an optimal solution if one exists, but they cannot reach the performance requirements of real-time systems. As a consequence, they do not scale well, and they also cannot handle requirements of dynamic environments. The main approach to the problem of finding the most effective organizational design has been to generate and search. Horling [21] has proven the problem of finding a valid organizational design to be NEXP-Complete. He has also shown that a knowledgeable organizational design significantly reduces the exploration effort. The heuristic methods do not guarantee finding an optimal solution, but they do guarantee finding a solution. In this work we will rely on satisficing solutions, and heuristic methods and take environmental factors into consideration for finding an effective organizational design.

The next section provides an overview of the related work on the heuristic models.

Heuristic Organization Selection

Applying a series of constraints to a system can help reduce the number of candidate organizations. This can be helpful to reduce the search scope. There have been various research attempts trying to apply constraints in order to facilitate the decision making process. Knowing that any heterogeneity of agents or their ability to apply multiple roles, increases the complexity of the system exponentially, some researchers have tried to enforce some level of homogeneity on the systems. Horling [21] achieves such constraints by introducing a homogeneity model and an abstraction model on a system called KB-ORG. The homogeneity model reduces the number of decisions for organizational choices through enforcing some amount of similarity as design time constraints. An example is enforcing the same model of aggregators in an information retrieval domain. In an abstraction model, it is tried to simplify the elements of the structure by removing any unnecessary or optional details, or by capturing them with a probabilistic model. These homogeneity and abstraction models can reduce the expected gain of an organization, as they can be very limiting or probabilistic.

3.2.4 Reorganization

The concept of reorganization can be applied to various aspects of a multi-agent system. A reorganization includes changes in any of the following aspects:

- *Number of Agents* - An open multi-agent system can have a variable number of agents by supporting agents leaving and joining the organization.
- *Properties of Agents* - The agents in a multi-agent system could possess learning skills that let them obtain new skills over time. The system could also possess properties that let the agents lose old skills and abilities.

- *Roles and Relationships of Agents* - The structure of an organization can change by means of changing the roles that various agents have, and how they interact with the other agents in the system.

The problem of reorganization in Multi-Agent Systems (MAS) has several aspects to it. The main problem in reorganizing a MAS is *the definition of the criteria to evaluate and find the most promising organization to reorganize to*. Other problems in reorganization are handling commitment issues of the individual agents that were committed to tasks or subtasks before the reorganization, without causing much drawback. Different application domains have their own sets of problems which have lead to solutions specific for them such as case based reasoning, learning, negotiation, etc. To our knowledge, there is no domain-independent reorganization model designed so far that has an acceptable performance for real-time systems.

At the same time, reorganization can be carried out in several ways:

- *Controlled (top-down)* -The reorganization is carried out as a known process. This could be performed for example by means of an expert system which controls the reorganization. There have been two main approaches to a controlled reorganization: (1) Exogenous approaches which let the MAS user control the reorganization process [60], and (2) Endogenous approaches in which the system will carry out the reorganization either by a decentralized (several or all agents involved) or centralized (a central agent involved) method. In a controlled model, the reorganization is initiated when it is deemed necessary and the system does not know when it will reorganize.

- *Emergent (bottom-up)* - There is not any kind of explicit control on the reorganization. The reorganization happens through implicit interaction of agents that have their own methods. The main problem of emergent techniques is the time it takes for the system to unfold. This can be a major problem for real-time systems.

- *Predefined* - The reorganisation is already planned and is expressed, for example, as a temporal organization model [9] . For instance, a soccer team has previously accorded to change its formation after 30 minutes of the match.

Reorganization Overhead

Every reorganization will have additional computation and communication costs associated with it. Depending on the type of reorganization, the origination of overhead costs can be different. As an example, in a centralized model of reorganization, the overhead costs will be associated with the organization manager. As a result, there needs to be a balance in how often an organization is reorganized.

3.2.5 Related Work on Reorganization

One of the earliest approaches to reorganization by means of cooperative agents relies on agent composition and decomposition. Ishida *et al.* [31] employ a reorganization trigger that is based on statistics from the organization's and the agents' performances. An agent is not aware of any other agent's statistics. But each agent knows how the whole organization is performing based on the organization's statistics. The agent statistics and performances are based on how busy an agent is over a specified period of time. The performance of organization is based on a predefined time limit of the task ($T_{deadline}$) and the most recently observed response time ($T_{response}$). If $T_{deadline} < T_{response}$ and an agent is completely busy, a decomposition is performed. If an agent is deemed idle, a composition is performed. A composition is also performed if the $T_{deadline} > T_{response}$ and an agent's performance is lower than $T_{deadline}/2T_{response}$. In this work we focus on reorganization as structural adaptation and role changes instead of role compositions and decompositions.

Another model of a cooperative reorganizing system is the Adaptive Multi-Agent Systems (AMAS) theory [46]. This theory is based on the agents' awareness of Non-Cooperative Situations

(NCS) that are adverse to the organization. Three types of NCS are considered: (1) Signals from the environment are incomprehensible, (2) The perceived information from the environment does not initiate any activity in the agent, and (3) The conclusions are not useful to others. If an agent deems itself as non-cooperative, it tries to take actions through its decision mechanisms to return to a cooperative situation. A specific list of NCS needs to be defined by the designer at design time. This limitation makes the AMAS theory non-applicable for our goals, because that approach can only be applied to environments in which all the states of the organization and environment can be identified at design time, which is not true in this work.

Hubner *et al.* [27] proposed a decentralized model of controlled reorganization. They suggest a reorganization trigger that is solely based on the performance level of the current organization. In this work, we employ an active reorganization trigger that is not only based on system performance, but also it takes into consideration changes in task type. At the same time, their system requires application specific implementation of several agents that handle the reorganization. Those agents include *monitor agents* to decide when a reorganization is needed, *selector agents* to select the best organization to switch to, and *designer agents* to manage change to another organization structure. Also in their work the organizational space is limited by a set of application-specific, hard-coded organizational preferences attained by the set of designer agents. For example in an application of small size robot soccer league, one designer agent ”always sees a plan to change the current organizational structure to a new one where the players' area is increased.”

Dignum *et al.* [11] discuss reorganization in agent organizations by classifying the various motivations for reorganization and various methods for reorganization. They broadly classify reorganization into two types: (1) behavior change involving short term behavior modification of some agents, and (2) structural change involving long term changes in the structure of the organization. Moreover, they emphasise the necessity of concretely determining the complete utility of an organization and its structure. Thus, while their suggestions further justify our proposed organizational performance evaluation method, they do not indicate any possible solutions.

Gaston and desJardins [19] define an Agent Organized Network as an organizational network structure that is the result of local rewiring decisions made by the individual agents in a networked multi-agent system. Gaston and desJardins propose the AON as an appropriate strategy for dynamic environments in which the agents only have access to local and uncertain information. But in their work, this strategy has been only applied to an initial arbitrary network topology in which the number of connections in the topology remains constant and any adaptation occurs only through one-for-one addition or removal of connections (rewiring). They are also considering a constant number of skills assigned to each agent, and a constant number of skills required per task, and the same utility for all tasks. Still their work shows that the adaptive network structure almost doubles the organizational performance after some iterations.

Ghijsen *et al.* [20] address the issue of designing agents in dynamic organizations. They introduce AgentCoRe which provides a framework for agent coordination and reorganization by means of a set of decision making modules for agents. These modules enable the agents to make decisions about dynamic selection of coordination mechanisms, task decomposition, task assignment, and adaptation of the organizational structure. AgentCoRe employs domain specific procedural descriptions as a set of strategies for decomposition, assignment, and reorganization. The reorganization decision is a function of task assignments. For instance, in a RoboCupRescue simulation, the trigger fires if there is an agent that has not been assigned a task and if there is at least one task that is still being executed.

Kota *et al.* [32] rely on inefficient patterns of communication as a trigger for reorganization. For example, if one agent is the center of a task and all the other agents coordinate via this agent, their proposed system detects this inefficiency and creates a direct communication. For example, if agent x keeps sending messages to agent z via agent y, then the system will eventually create a direct communication between x and z. As a result, their system might not detect all possible cases that a reorganization might be advantageous. For example, in a target detection and tracking application which employs coalitions to detect and track objects, their system will never find any pattern of inefficient communication. Because in such an application, there is a continuous change in coalitions of three agents to a new coalition with three new agents. So their system is limited to applications that follow some pattern in the communication of agents, which might not be the case for dynamic systems in open environments. At the same time, some of the methods make the assumption that all agents are acquainted with each other. This assumption is limiting as it places requirements on abilities of agents and at the same time, puts limitations on the whole system in case of open worlds as it requires global updates for all the agents that are in the system.

Most of the other applied reorganization models, rely on reorganizing using the same organizational structure. Zhang *et al.* [63] employ a self-organization model for coordinating decentralized reinforcement learning using a hierarchical structure. In their work, the agents dynamically reform to different hierarchies. That approach is selected to reduce the complexity and increase the speed by which the system reaches a convergence.

Durfee and Montgomery [15] have used team-level abstraction in order to reduce complexity and to leave some specific agent assignments unbound during coordination. This team-level abstraction also reduces the precision by losing the details that were previously stored within individual agent nodes. At the same time, their model relies on a static organizational structure that is teams.

Barton and Allen [3] develop a task selection strategy for agents which helps the system to reach a solution quicker. They set a strategic behavior for agents by giving them preference to join an existing task over rewiring, proposing, or waiting. But their work is based on a single organizational structure that is a coalition.

In a similar approach, Singh *et al.* [55] rely on reorganizations within a coalition structure. Smart *et al.* [56] compare the performance of dynamic and static coalition structures. Their study proves the dynamic model of coalition structure to gain a higher performance.

In summary, the related work on reorganization in multi-agent systems is limited to changes in one type of organization structure, or is just about change of agent roles and not relations.

Most systems are also limited by requiring a set of hard-coded, application or domain-specific organizational preferences and procedures attained by the designer of the system at the design time. At the same time, most of the systems are only applicable to static and closed-world environments by requiring all states of the application and environment to be identified at design time. Also, these models fail to identify all cases that a reorganization can be beneficial.

In relation to the related work on heuristic solutions to organization selection, most of the systems rely on one static organizational structure. At the same time, some of the proposed systems apply abstraction or probabilistic models which not only limit the expected gain of the system, but also make the model application-specific.

In this work, we employed a structural reorganization model that enables the system to reorganize between different types of organizational structure. Instead of attempting to discover non-cooperative or low-performing situations, we employ a task-based model for triggering reorganization. This task-based model together with a multi-level control structure reduce the costs associated to reorganization and thus make feasible a real-time reorganization method which can enhance the overall performance. Our methodology does not rely on a central monitor agent to determine when reorganization is necessary, instead individual agents are enabled to contribute to reorganization.

Chapter 4

Research Methodology

In this work, we employ an explicit organizational model to enforce organizational constraints to the system. We acheive this by explicitly adding an organizational layer to the system. Figure 4.1 presents how the organizational layer is added to the whole system. The following sections provide more details about our approach to this problem.

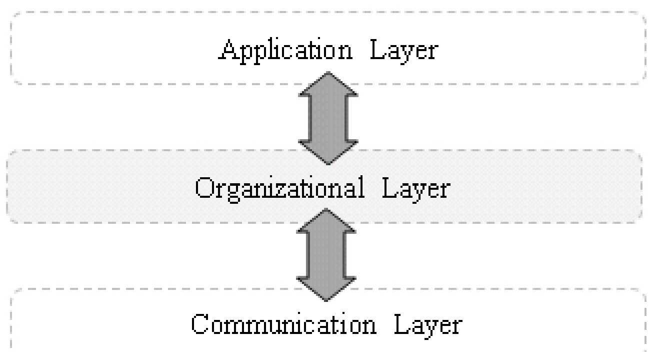


Figure 4.1: Adding an organizational layer

4.1 Framework for Structural Reorganization

The gains that an organization can attain are largely predetermined by means of the value potentials created beforehand [50]. Value potentials are defined as the set of rules and prerequisites that if fulfilled, the organization can provide certain benefits. Resources, capabilities, and core competencies are samples for such prerequisites. We use strategies to enforce value potentials in an organization, and focus on strategy formulation in multi-agent systems as a means to adapt to changing goals, tasks, and resources in dynamic environments. We define a *strategy* as a plan of action that lets the system decide how it allocates resources and roles to the agents, how the agents interact and cooperate for a specific task type, and based on conditions of the environment and characteristics of the current task. In other words, a strategy is a plan of action for determining what organizational structure is used for each task type. We rely on the concepts that organizational structure and strategy are interrelated and a successful organization requires establishing this relation in making organizational control decisions [52]. Strategic behavior should be based on a dynamic model for strategy formation which takes situational factors into consideration, such as crisis/no crisis in performance, presence/absence of a strategic micro-structure, power dependency of inter-organizational relations, goals and resources of the system, etc.

An important aspect of any organization is its ability to evolve over time. As the environment, goals, and individuals evolve, an organization should try to adapt to new conditions by altering patterns of interaction among its constituent agents. Adaptive organizations have the chance to achieve coherence in open and changing environments [59]. Organizational adaptation becomes critical especially in dynamic environments. A reorganizing behavior can facilitate properties of some applications that are too complex to have a priori algorithm, or that are linked to real world and open environments (e.g., the Internet), and do not have a fixed best design guaranteed. Bernon *et al.* [4] provide a comprehensive overview of several examples of applications that have been benefiting from the reorganizing behavior of agents in a multi-agent system. It should be considered that adaptive agent behaviors do not always lead to an adaptive organizational behavior [7]. Gaining an integrative view of an organization is an essential means to effectively managing changes in an organization.

Following Schwaninger's model of systemic control, we employ a multi-level control structure using two levels of control referred to as *strategic management* and *operative management*, which are used for making organizational control decisions. The strategic management layer is used for making higher level structural change decisions. This layer of control relies on having a categorization of task types and their requirements, and also creating a relation between the organizational structure that would be more appropriate for the requirements. With occurrence of any changes in goals of agents, this layer of control is used for determining what is the most effective organizational structure to be used. The operative management control allows the individual agents to make operative control decisions. An interrelationship between these levels of control makes it possible for the higher level of control to exert a pre-control influence on the lower level. As a result, the collective actions of agents in operative management layer are affected by the selected strategic behavior on the strategic management level.

4.2 Modeling Organizations and Reorganizations

Most models that support explicit representation of an organization, focus either on the functioning or structure of the organization. The MOISE+ (Model of Organization for multIagent SystEms) [26] framework provides a rather complete infrastructure for modeling organizations by providing a single, coherent way for modeling both the function and structure of an organization. Figure 4.2 presents how an organization with both functional and structural aspects affects the agents' behavior by explaining or limiting their behavior space. In this figure, it is supposed that the agents try to maintain their behavior in space G, where G presents all behaviors that satisfy the agent's current goal. Space E presents all possible behaviors in the current environment. Space S presents all agent behaviors which satisfy the requirements specified by the organizational structure, such as roles, relations, group formations, etc. The agents try to get their possible behaviors $(E \cap S)$ closer to G, thus avoiding the $(E \cap S) - G$ space. The organizational functioning space contains a set of behaviors that have been proven effective for turning the agents behavior towards the G space.

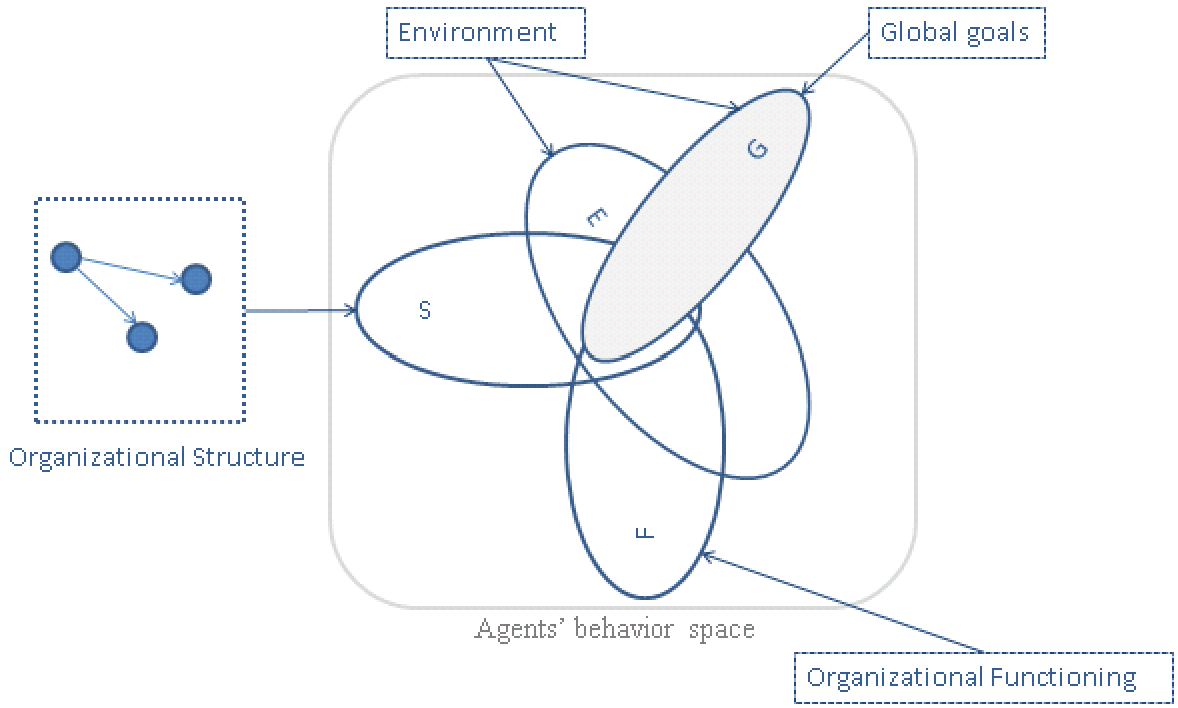


Figure 4.2: Organization effects on agents' behavior

Figure 4.3 illustrates the changes in relations using our proposed reorganization model. As the figure shows, more than one organizational structure is made available to the agents. This figure also shows how the functional aspect of the organization is affected by the choice of the structural aspect and how the global goals and environmental conditions have a direct effect on the organizational structure.

Using the MOISE+ framework, an Organizational Structure (OS) is formed based on definition of the structural, functional, and deontic dimensions. The structural aspect is used for defining the agents' relations based on the concepts of roles, groups, and links. The functional aspect is used for determining how the organization achieves its global goals; this includes goal decomposition(plans) and task distribution(missions). Global goals, plans, and missions are specified using a social scheme. Role permissions and the mission obligations are specified by the deontic aspect.

When a set of agents adopts an OS, an Organizational Entity (OE) is formed. Once an OE is formed, its history starts and is composed of events such as agents entering or leaving the organi-

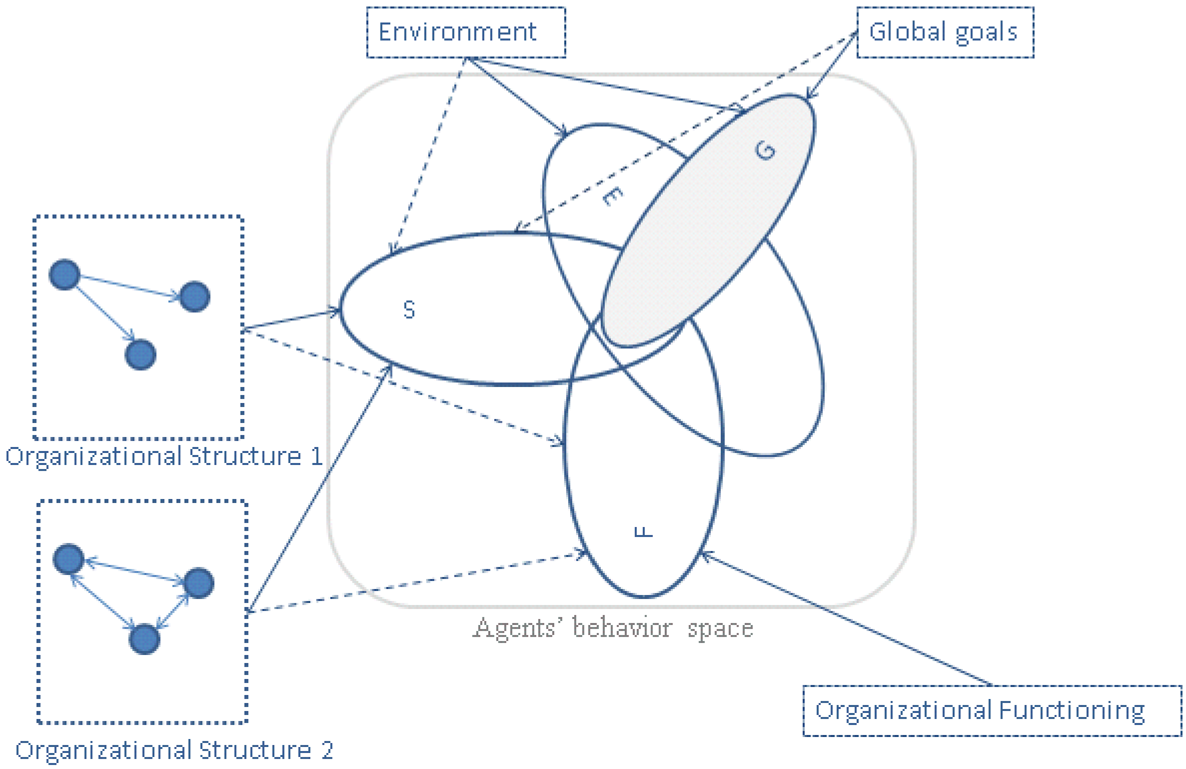


Figure 4.3: Reorganization effects on agents' behavior

zation, role adoptions, mission commitments, group formation and deformations, etc. A reorganization is therefore a change in either the OS or the OE. In our proposed reorganization model, reorganization is performed as a change in the OE at run time, while the OS is written to accomodate for such a potential change of the OE by having access to the structural specification of all the potential structural models to which the agents can potentially reorganize.

Figure 4.4 presents a more detailed look at the organizational layer added to the system using different classes from the MOISE+ framework. A special agent called OrgManager maintains the state of the current OE. The agents can then send messages to the OrgManager using their OrgBox API. The agents can ask for organizational events such as group formation or deformation, scheme creation, role adoptions, mission commitments or de-commitments. Broadcast of message to group members are sample events that an agent can generate. It is the responsibility of the OrgManager agent to make sure what the agent has requsted does not contradict the organizational specification.

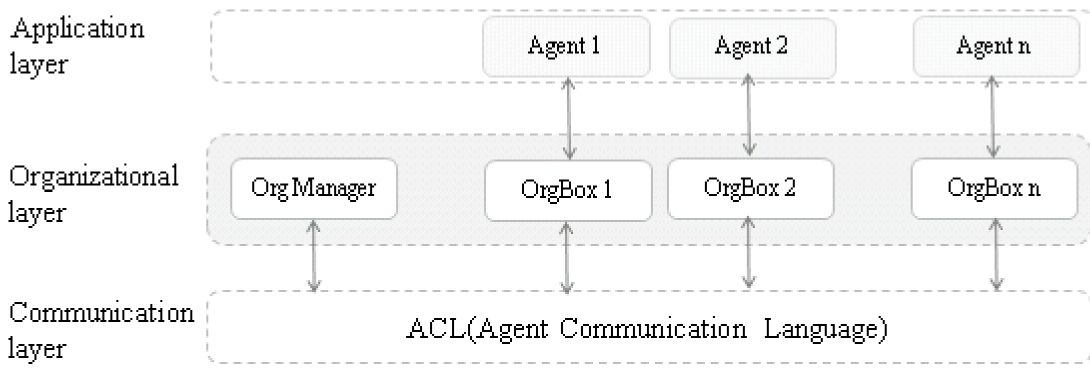


Figure 4.4: Component interaction using MOISE+ organizational layer

Overall, MOISE+ allows specifiying three levels of structural behavior: Individual, Social, and Collective. Further clarification of each of these levels and how MOISE+ supports them follows:

- **Individual level** - This level is formed by the roles of individual agents. A role can be described as a set of constraints that an agent accepts when joining the OE or a group. Role constraints are defined in two ways, in relation to other roles, and in a deontic relation to global plans. The roles can have inheritance relations which allows one role to be a sub-role or a specialization of a more general role.

- **Social level** - This role is formed by means of organizational links which define the agent relations. Three types of organization links are defined: communication, authority, and acquaintance. A communication link specifies the kind of communication that can exist between the roles, the protocols that have to be followed, and any particular missions for which they can be used. The subordination of roles is determined by use of the authority links. The authority links also specify the context within which the subordinate relation is valid. The context is defined by means of the missions that are associated with the link. All the roles about which an agent can possess information and can use in its decision making mechanism are specified by acquaintance links of a role.

- **Collective level** - This role is formed by means of compatibility constraints that specify what roles an agent can play depending on the agent's current role. This level is also responsible for specifying group formation constraints and requirements. A group is composed of a set of roles, missions, and the links that exist between the roles which belong to the group. Each group specification clarifies various roles that are needed to form this group and the cardinality of each role. The groups can also have inter-group and intra-group compatibility relations. This makes possible formation of subgroups. A group is considered well-formed if it conforms with both role and sub-group cardinalities.

In a higher level view, the organizational structure in MOISE+ is modeled in form of a graph that is defined as a set of roles, links, and groups. Each role is composed of a set of missions. A mission is a permitted behavior in the system, defined by a set of goals, actions, plans, and resources. Assignment of a role to an agent causes the agent to follow the permitted behaviors specified by the missions of that role. The interaction between the roles is specified by means of the organization links.

The functional structure in MOISE+ is based on the concept of missions and schemes. A scheme is basically a goal decomposition tree with the root being a global goal and missions being the responsibilities for the sub-goals. A mission is a set of coherent goals that an agent can commit to. If agent a1 accepts mission m1 which has two goals $\{g1, g2\}$, then by accepting m1, the agent has committed to g1 and g2. An agent will try to acheive a goal only when the preconditions for that goal are met. Similar to groups, a mission is considered well-formed when the number of agents that have committed to the mission reaches the minum cardinality specified for the mission. A preference order can be specified between missions, thus if an agent has committed to two missions, it will know which goals it has to exceute first.

In order to develop a relation between the structural and functional aspects of an organization, a deontic specification is used for specifying any such relations. In MOISE+ these relations are specified in the individual level using permissions and obligations of a role on a mission. In the context of Figure 4.2, the deontic specification delimits the set $S \cap F$ among the set S (the allowed behaviors).

4.3 Evaluating Effectiveness of the Proposed Methodology

One of the main application areas that benefit from multi-agent architecture focuses on solutions that try to efficiently use information sources that are geographically sparse. Sensor nets, seismic monitoring, and information gathering from Internet are some example applications in this area. To evaluate the efficacy of our proposed methodology, we have developed two different simulations using scenarios that are representatives of geographically dispersed cooperative distributed problem solving applications. A brief description of each of these applications follows:

- **Pursuit game** - The general scenario follows the typical pursuit game in which a group of predators try to catch preys in a grid world. In an enhanced model of pursuit game, we limit the field of view (sensing radius) for predators to a certain variable. This variable can be set at the beginning of an experiment run. In a typical application of a pursuit game, the predators have the ability to observe the whole world. A successful detection of a prey requires a sensory detection confirmed by at least two predators, while in a typical pursuit game such a requirement is not enforced. At the same time, the sensory input of a prey is not limited, meaning that preys can observe the whole world and become aware of the location of all existing predators. We also enhance the prey movement by an algorithm that enables the preys to move away from detected predators. It is very common for a prey movement algorithm to follow one specific pattern, for example a diagonal or straight line.Also, a typical pursuit game is composed of 4 predators and 1 prey, while we enabled the system to have any number of predators and preys. We use different numbers of predators and preys in our experiments. Chapter 5 provides further details about this application.

- **Cow herding** - The scenario for this application is taken from the 2010 Multi-agent Programming Contest [1]. In this scenario, two teams of agents compete by trying to herd cows into their own corral. The cows are spread throughout the world either in herds or as dispersed. The task and world complexity can be different depending on the world model provided by the simulation server. In more complex world models, additions of fences and gates increase the task complexity. The starting positions of the agents are determined by the world models. In some models, the agents are spread throughout the world, while in other models the agents get started in one area and close to each other. All these environmental factors affect the overall complexity. Further details of this application are discussed in chapter 6.

We enhance each of the above applications with an explicit organization and reorganization model by using S-MOISE+ and J-MOISE+ which are two APIs based on the MOISE+ framework. These variant APIs enable us to apply our methodology to two systems with two different agent architectures. S-MOISE+ relies on a reactive agent model and uses SACI as the agent communication infrastructure. J-MOISE+ relies on a BDI agent architecture and can be used in conjunction with agents written in JASON language. Next two chapters report on implementation details of each of these experimental setups, evaluation of their performance, and results of the experiments.

Chapter 5

Case Study Using a Pursuit Game Simulation

Multi-agent systems have been a common platform for investigating distributed cooperative problem solving. We are specifically considering a domain in which resources are limited and agent cooperation is required for a successful task completion. This chapter provides an introduction to the basic scenario of the pursuit game. A modified model of the pursuit game is then described. The organizational modeling framework used for this application is elaborated on together with discussions of the implemented organization and reorganization models. Experimental setup, experimental runs, and results of the exeperiments are presented. The chapter includes the analysis of the results.

5.1 Scenario

The pursuit game has been a popular application domain to address cooperative behavior in multi-agent systems. Despite its simplicity, this game model can provide a rough abstraction of more complex real-world scenarios. In the most basic and common scenario, four agents play the role of predators and try to catch an escaping prey agent. The prey is considered caught when it is surrounded by a combination of predators and world boundaries in a way that it cannot move any more. In different models of the game, the predators and preys can follow different behavioral models. For example, the prey can move in a straight or diagonal line, or it can act more intelligently by moving away from predators. None of the predator agents, regardless of their skills and intelligence level, can accomplish the task of catching a prey without cooperation with other agents.

In this work, we use a modified model of pursuit game to represent a sample distributed problem solving application. We add certain enhancements to the population of agents and apply certain limitations to capabilities of predator agents and the task requirements. The goal is to increase complexity of the game and resemble a multi-object detection and tracking application in a distributed sensor network establishment. These enhancements and limitations are:

- The preys are aware of the location of predators and move in a direction that maximizes their distance from the predators.
- The predator agents are limited to sensing a certain radius around them.
- Each predator has one of four possible skills.
- Agent skills do not change over time.
- A successful detection of a prey requires sensory data from more than one predator agent.

5.2 System Model

We developed a limited model of a problem solving agent organization with the following specifications:

- The organization is closed, that is no agents can join nor leave the organization once the organization is running.
- The agents are invariant, that is their properties do not change over time.

- Environmental conditions stay the same.
- Task type changes for agents over time.
- Agent relations change over time.
- Agents are only self-aware. They have no global view of the world, neither any acquaintance with other agents that might exist in the system. Unless an agent becomes a member of a group, in which case it will form relations with other agents in the group.

5.2.1 Agent Model

Three types of agents are modeled. A WorldAg which represents the world and is modeled as an n by n grid with boundaries, but no obstacles. The WorldAg implements the world behavior by maintaining the position of agents and providing the position information to any agent that might ask for it. The world gets populated with two types of agents,

- Predator agents, $A1 = h1, ..., h|N|$, have the capability to sense a radius of 4 around them. Each predator agent has one of four possible skills. An Up-predator tries to position itself above the prey that it is trying to catch, a Down-predator tries to position itself below, Right-predator to the right, and Left-predator to the left of the prey.
- Prey agents, $A2 = p1, ..., p|Q|$, are aware of the location of predators, and try to maximize their distance from the predators.

Both the Predator and Prey agents have a reactive architecture based on which they follow a stimulus-response behavior without maintaining any internal state. As these agents do not have any representations of their environment, they can only take local information into account. The agents can form acquaintance, communication, or authority relations. Each role spans the properties of the previous role in this list. In an acquaintance relation, the agents are only aware of the existence

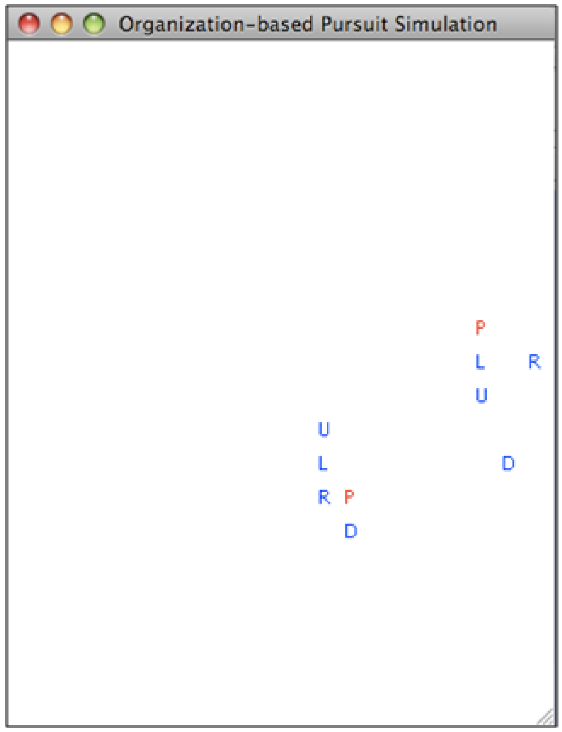


Figure 5.1: Sample world populated with 8 predators with sensory detection radius of 4

of each other, and have no communication relation. In a communication relation, the agents are aware of the existence of each other and they can also communicate with each other. In an authority relation, the agents are aware of the existence of each other, can communicate with each other, and the agent that has the authority can determine actions of a follower agent.

In this model, the predator and prey agents have no relations at the beginning. The world agent has acquaintance relation with all predators and preys. Agent relations change over time. Once an agent becomes member of a group, then it forms a relation with other group members. The type of this relation depends on the type of group and the agent's role in that group. The world gets populated with a random placement of predators and preys on the grid. Figure 5.1 presents a graphical view of a sample game. In this example, the world is populated with two prey agents which are represented by the letter P, and eight predator agents that are represented by letters U, D, L, and R depending on their skills.

5.2.2 Cooperation and Task Decomposition in a Pursuit Game

The predators only have access to local information, so it becomes necessary that they cooperate and share information. Also, because each predator has a specific skill, cooperation of predator with complementary skills becomes essential for successful task completion. The predators try to form groups or join other groups depending on their sensory input, their current task, and any requests from other predators. Each predator can only be member of one group at a time. A predator moves randomly if it has not detected any prey itself nor it is receiving any information from its group members. A prey is caught and removed from the world when it cannot move any more. A prey will not be able to move if it is surrounded by predators or by a combination of predators and the world boundaries. Each predator tries to move towards the prey and in a position that fits its skill.

Two types of tasks are identified in this simulation model. The first task is *detecting* a prey, and the second task is *tracking and catching* a detected prey. Both these tasks require agent collaboration. The prey detection task requires cooperation of at least two to four predator agents to confirm that they have sensed the prey. A successful completion of the track and catch task requires a combination of at least two to four predator agents with different skills.

When a predator senses a prey, it tries to form a group with other predator. Predators try to form groups with the predators that are closest to them in proximity. Once a group is formed, agents in the group follow their roles and relations, as specified in the structural specification, to share information about preys. A well-formed group of predators will have one skill from each of the four possible skills. No group can have more than one predator agent from each skill. The organizational framework ensures that group formation follows the specified requirements. Once a group of agents achieves its goal, it deforms or reorganizes to a different type of group.

5.2.3 Organization and Reorganization Models using S-MOISE+

We determined that S-MOISE+ [28] was suitable for designing our problem solving agent organization.This framework enables us to make sure that the agents within the system follow the organizational constraints. Using this framework, the organization is interpreted at run time and it is not hard-wired in the agents. S-MOISE+ is an open source implementation of an organizational middleware that follows the MOISE+ specification. This middleware acts as the interface between the agents and the overall system. Using this middleware, agents get access to the communication layer, they can get updated information about the current status of organization such as formed groups, generated schemes, role assignments, etc. The agents also get the ability to affect and change the Organizational Entity (OE) by means of the middleware.

The MOISE+ framework introduces an organization by considering three dimensions into it. A structural dimension defines roles and links of inheritance and groups; a functional dimension specifies a set of global plans and missions for the goals to be achieved; and a deontic dimension is responsible for assigning obligations and permissions to roles with respect to missions. S-MOISE+ employs an OrgManager agent that has the current state of the organization and keeps it consistent. Each agent in the system uses an OrgBox API to access the organizational layer. Using the S-MOISE+ framework, an Organizational Specification (OS) is defined which contains the static description of the organization. This description covers all different aspects of an organizational model such as types of roles and the behavior that an agent is responsible for once it adopts a role, interconnection between roles, groups, collective goals, and how the goals are decomposed into plans and distributed between agents. The current instance of an OS is referred to as an Organizational Entity (OE). An OE is formed by a set of agents adopting an OS and thus having a common goal. In this work, we consider reorganization as a change in the OE.

The S-MOISE+ API has three main classes for accessing the organizational layer:

- OEAgent - Represnts the agent within the organization, and stores the agent's roles and responsibilities.
- OrgBox - Has methods for creating organizational events such as group creation or commitment to missions.
- BaseOrgAgent - Implements a general agent architecture which enables an agent to iteratively choose a goal, make a plan, execute the plan, and then loop back.

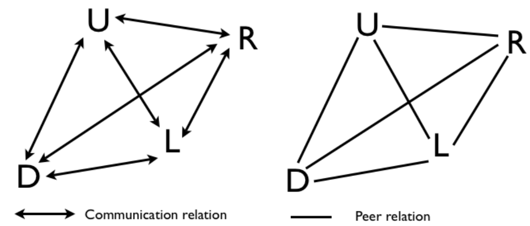
The OrgManager ensures that the organizational events which are generated by the agents adhere to the following organizational constraints specified by MOISE+:

- Maximum cardinality of roles in a group
- Role compatibilities
- Commitment only to permitted missions or obligated roles
- Creation of roles, groups, and schemes only based on the original specification

Using the S-MOISE+ framework, we have implemented two different models of organizational structure and a reorganization model. A short description of each of these models and the agent interactions within each follows.

Model A - Coalition Structure

We consider coalition as a goal-directed and short-lived group of agents that is formed with a goal in mind and dissolves as soon as the goal is satisfied [21]. All agents in a coalition are peers and have a communication relation. The agent that initiates formation of a coalition, acts as the representative for the coalition. Figure 5.2 illustrates communication and authority relations of four predator agents in a coalition. In a simulation model that only uses the coalition structure, coalitions of agents are used for performing any task that requires group work. In this case, both



(a) Communication relations in a coalition

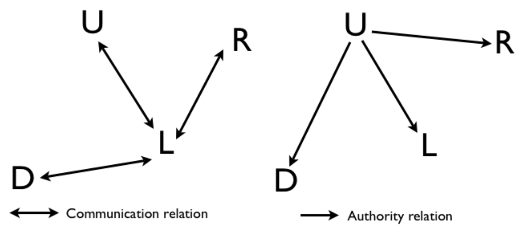
(b) Authority relations in a coalition

Figure 5.2: Agent relations in a coalition structure

detecting and tracking a prey are performed using coalitions of agents. Using this model, each agent relies on its own sensory information if it senses a prey, and shares that information with the rest of group members if it is part of a group. If the agent is not part of a group, then it tries to form a group with the closest agents. If the agent does not sense a prey and it is part of a group, then it looks to use any shared information. If no shared information is available either, then the agent moves randomly.

Model B - Team Structure

We consider a team as a number of cooperative agents that coordinate to be supportive of the team's goals [21]. In this work, each team will have a leader that maintains an authority relation with all the other team members. Figure 5.3 demonstrates agent relations in a team with agent U as the leader. Similar to a coalition model, in a simulation model that only employs a team structure, teams of agents are used for performing any task that requires group work. In this case, a team structure is used both to detect, track and catch preys. The major difference in agent's behavior between team and coalition is that only the team leader agent can share information with the rest of group members. Also, if a team member has received information from the team lead, then it will just use the shared information instead of using its own sensory information.



(a) Communication relations in a team (b) Authority relations in a team

Figure 5.3: Agent relations in a team structure

Model C - Dynamic Organization Structure

Certain characteristics of various tasks make them more appropriate to be handled with a certain organizational structure. For example, while coalition might be the most appropriate structure in terms of time efficiency for detecting a moving object quickly, a team with a leader might be the best structure in terms of resource efficiency to track a moving object once an object is already detected. The dynamic organization model enables the agents to use either of the team or coalition structures and also to be able to reorganize between the two.We consider reorganization as a change in the current Organizational Entity (OE) which makes it possible to have a structural change between coalition and team. As part of the strategic management of the organization, we employ a task-based reorganization model. Using this model, the current task type is used for determining the organizational structure to be used by a group of agents.

Figure 5.4 illustrates the authority and communication relations of agents in the sample world populated with eight predators and two preys. One group of agents is using a coalition structure, and the other group is using a team structure.

The predators try to adopt a CatchPreyScheme which provides the agents with the set of goals, as illustrated in Figure 5.5. Each team adopts these scheme and goals with its own structural model.

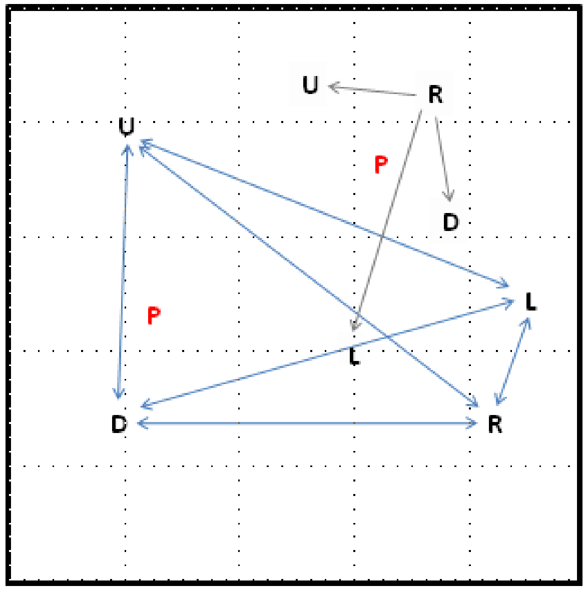


Figure 5.4: Authority and communication relations in a sample pursuit game

5.3 Evaluation Criteria

The effectiveness of each model is measured in terms of successful task completion and the associated cost. The world gets populated with a fixed number of preys. The goal of predators is catching all the preys in a limited amount of time. The success rate of each experiment is measured using the number of preys that are caught and the time it takes to do so.

$$SuccessRate = Sum(\frac{\sum_{x=1}^{|Q|} caught_{P_x}}{|Q|} * \frac{totalTime}{elapsedTime}) \tag{5.3.1}$$

The cost associated with each agent in the organization is measured both in terms of number of messages passed, and number of moves that the agent has to make. The communication cost also includes any messages used for reorganization. The total communication cost is measured as,

$$CommCost = C.\sum_{x=1}^{|N|} c_x + R.\sum_{x=1}^{|N|} ReOrgCost_x \tag{5.3.2}$$

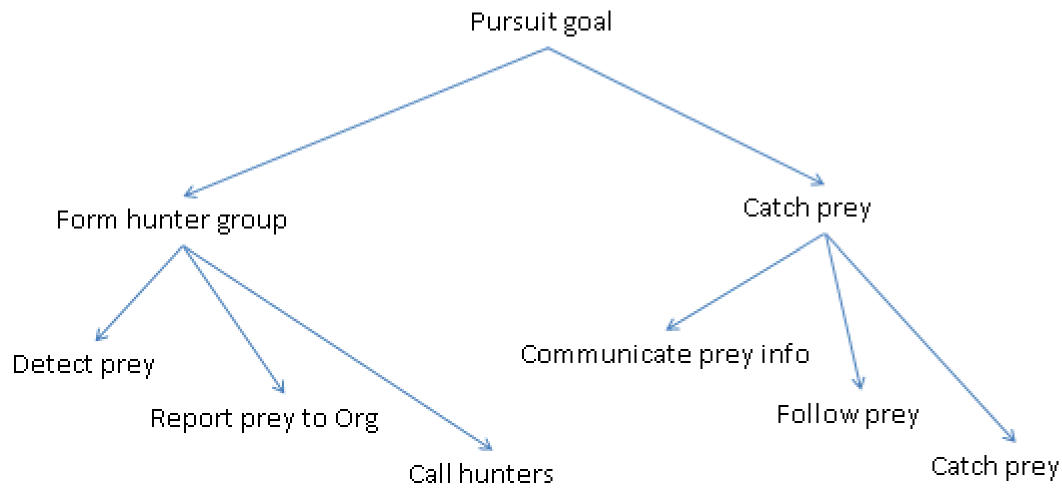


Figure 5.5: Goal decomposition tree in the pursuit game

where C is the communication cost coefficient, c_x is the number of messages sent by agent x, R is the reorganization cost coefficient, and $ReOrgCost_x$ is any overhead messages passed for reorganization by agent x. All messages are considered to have the same cost regardless of their size or travel distance. The total move cost of predator agents is measured as

$$MoveCost = M.\sum_{x=1}^{|N|} m_x \tag{5.3.3}$$

where M is the move cost coefficient, and m_x is the number of moves for that agent.

The total cost is a combined measure of the total communication and move costs,

$$TotalCost = C.\sum_{x=1}^{|N|} c_x + R.\sum_{x=1}^{|N|} ReOrgCost_x + M.\sum_{x=1}^{|N|} m_x \tag{5.3.4}$$

5.4 Experimental Evaluation

We conducted a number of experiments in order to evaluate the effectiveness of our task-based re-organization for different population of agents. The variables and parameters for the world settings are set as:

- The world size (grid size) is set to 20 by 20.
- The sensing radius for predators is set to 4.
- The world is populated with three different population sizes for predators and preys.
  - 4 predators, 5 preys
  - 8 predators, 5 preys
  - 12 predators, 5 preys
- The world gets populated with a random placement of predators and preys on the grid.

Using each of the three organizational models and by using three different populations of agents, an overall of 27 experiment sets are run. We run a total of 75 simulations for each experiment set. Each experimental run is either limited to 120 seconds of run time, or the experiment is stopped when the goal is achieved, that is all the preys in the world are caught. The time it takes to catch all the preys is captured for use in performance evaluations.

When evaluating the performance, we set the values of M, C, and R (communication and move cost coefficients) to 1. This means that we consider the same coefficient for reorganization messages as any other messages. While the communication and move cost coefficients can be different in a real world application, we set both of them to 1 and look into the data for each of these individually as well as the sum.

5.5 Experiment Results

The results are presented in terms of graphs and tables presenting the average values for success rate, communication cost, move cost, and total cost for each population of agents and for each organizational structure. A better performance is reflected in terms of higher success rate and lower total cost. Table 5.1 and Figure 5.6 present the success rate for each experiment set. Table 5.2 and

Figure 5.7 demonstrate the total cost associated with each. The breakdowns of communication and move costs are presented in Tables 5.3 and 5.4, and Figures 5.8 and 5.9.

Organizational model	4 predators	8 predators	12 predators
Coalition	0.17	0.222	0.322
Team	0.102	0.233	0.328
Dynamic	0.117	0.251	0.365

Table 5.1: Success rate for each set of simulation runs

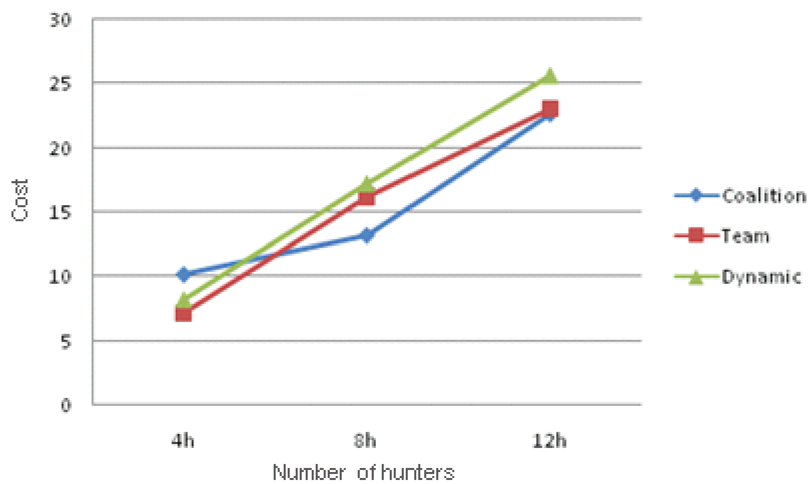


Figure 5.6: Success rate

5.6 Analysis

As the results demonstrate, the dynamic organization outperforms the coalition and team organizations once the agent population increases. This higher performance is achieved despite the additional overhead costs associated to reorganization. Also, with the increase in problem size,

Organizational model	4 predators	8 predators	12 predators
Coalition	17.995	21.868	25.509
Team	13.365	14.194	14.51
Dynamic	12.85	14.37	15.223

Table 5.2: Total cost for each set of simulation runs

Organizational model	4 predators	8 predators	12 predators
Coalition	14.25	16.701	25.509
Team	13.365	14.194	14.51
Dynamic	12.85	14.37	15.223

Table 5.3: Communication cost for each set of simulation runs

Organizational model	4 predators	8 predators	12 predators
Coalition	3.745	5.166	5.573
Team	4.349	4.735	4.919
Dynamic	4.183	4.835	5.116

Table 5.4: Move cost for each set of simulation runs

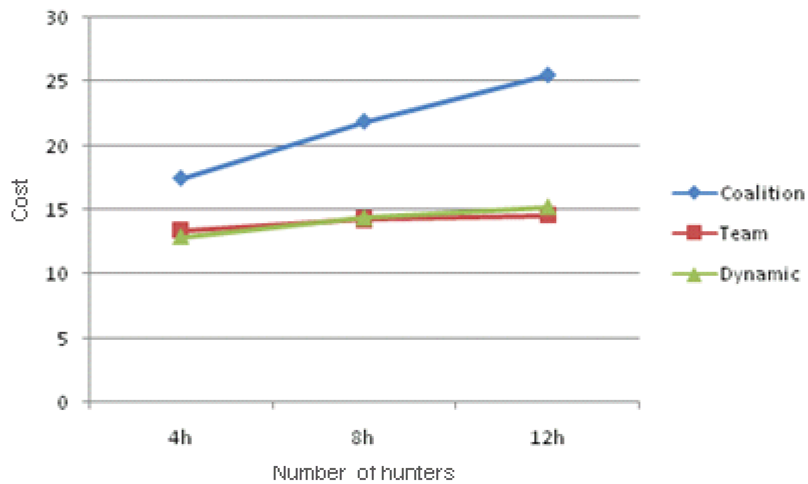


Figure 5.7: Total cost

the relative difference in performance of the dynamic organization with the other organizational models increases. This result demonstrates a relation between the problem size and effectiveness of reorganization, as is also the case for social organizations. At the same time, the coalition structure has the highest performance when the problem size is small. Its comparative performance is reduced when the problem size is increased. The higher performance of coalition for the small population of agents is achieved with a significant increase in communication cost. Overall, the results confirm how the dynamic organization can be used to stay ahead of the organizational change and attain a higher performance when the agent population increases.

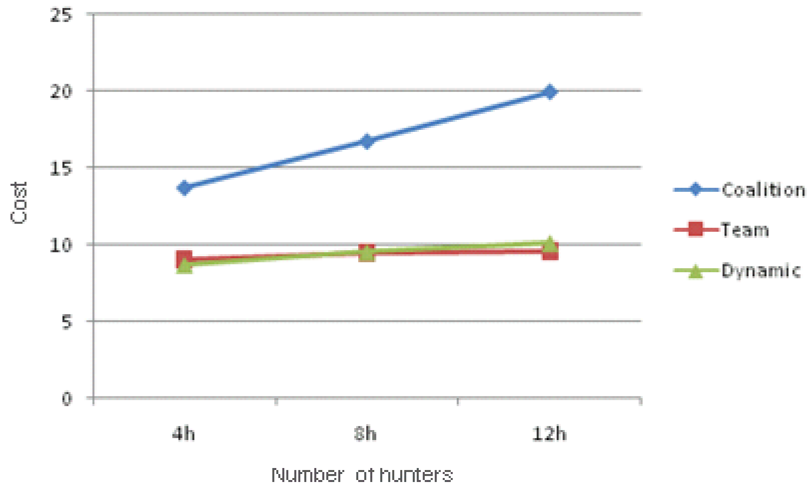


Figure 5.8: Communication cost

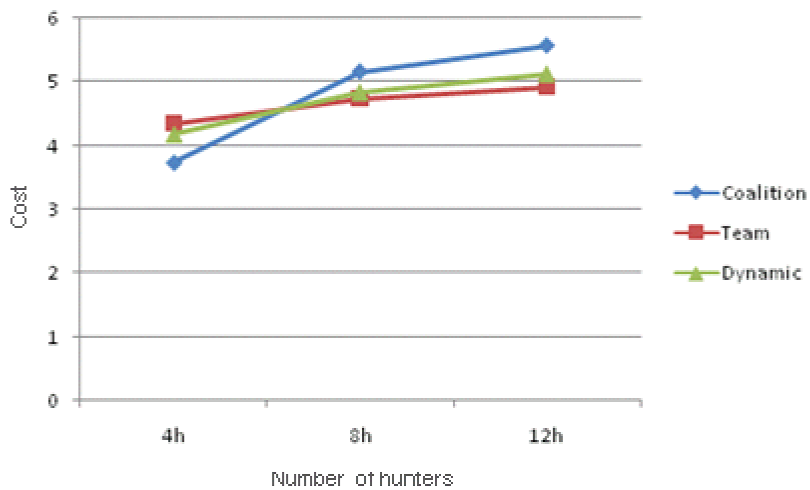


Figure 5.9: Move cost

Looking into the performance of dynamic organization and the team, the dynamic organization has a higher performance in all the experiments. While the communication and move cost between these two models stay comparably similar. This can be because of two factors. One is the very short period of time that the agents in the dynamic model spend as a coalition trying to detect a prey, compared to the amount of time they spend to keep track of the detected prey and to catch it. The other is the overhead costs associated with a dynamic organization which levels with the lower communication cost of the coalition.

Looking into the cost graphs, the coalition structure has the highest communication cost in all the experiments. This is an expected result because of the extensive peer communication between all the agents in a coalition. There is not a significant difference in move cost across the experiments. This indicates that none of the organizational models has affected the number of moves significantly.

In summary, we find that potential gains of a task-based reorganization in terms of success rate and lower cost can be attained when the problem size increases. Furthermore, for a small population of agents, using an effective static organizational model can offer a higher performance than a reorganization.

Chapter 6

Case Study Using a Cow Herding Simulation

An unbiased multi-agent evaluation depends on use of a reliable simulation unbiased by the specific agent design. In this work, we use a simulation server provided by the 2010 Multi-agent Programming Contest [1].

The multi-agent programming contest uses a cow herding scenario in which two teams of cooperative agents compete against each other for resources. This chapter provides a decsription of the scenario and the simulation model. J-MOISE+, the organizational modeling framework used for this application, is presented. The organization and reorganization models used in the simulation are described. Experimental setup, experimental runs, and results of the exeperiments are presented. The chapter includes an analysis of the results.

6.1 Scenario

”An unknown species of cattle was recently discovered in the unexplored at-lands of Lemuria. The cows have some nice features: their carbondioxyde- and methane-output is extremely low compared to the usual cattle and their beef and milk are of supreme quality and taste. These facts denitely catch the attention of the beef- and milk-industries. The government decides to allow the cows to be captured and bred by everyone who is interested and has the capabilities.

Several well-known companies decided to send in their personnel to the elds to catch as many of them as possible. This leads to an unprecedented rush for cows. To maximize their success the companies replace their traditional cowboys by artificial herders. In this contest the participants have to compete in an environmenr for cows. Each team controls a set of herders in order to direct the cows into their own corral. The team with the most cows in the corral at the end wins the match. [1]"

6.2 Simulation Model

The above scenario is simulated using a grid world with objects scattered on it that represent different entities. The grid size is a variable and is specified at the beginning of each experiment. The environment contains two corrals which are the places that each team should herd the cows into. Each grid cell can contain one of the following objects:

- Obstacle
- Fence
- Switch
- Cow - Each cow will have a unique identifier.
- Agent - The value of this attribute will have an indication whether this is an enemy agent or an ally.
- Corral - The value of this attribute will have an indication whether this is an enemy corral or an ally.
- Unknown - The contents of the cell is not provided by the simulation server because of information distortion.

Only one object can be in a cell, with the exception that Agents and Cows can enter cells which contain corrals.

6.2.1 Agents, Agent Perceptions and Actions

The agents of each team can play any of the following roles [?]. Agents will be able to switch roles as the simulation progresses and system conditions change.

- Explorer - Explores the environment until it detects a cow.
- Scouter - Follows the explorer
- Herder - Herds the cows until they get in the corral.
- Herdboy - Helps the herder to get cows in the corral.
- Gate keeper 1- Activates the switch on one side of the gate. The gate will not be opened until Gate keeper 2 activates the switch on the other side.
- Gate keeper 2 - Activates the switch on the other side of the gate. This agent has to stay by the switch to keep the gate open until all agents, from either group, who wanted to pass the fence at that time, have passed.
- Leader - An implicit role that either the explorer or herder can take on. This role will enable an agent to lead the other agents by providing information that affects their actions.

Agents only have a local view of the environment. The agents perception of the world can be incomplete, meaning that they can receive incomplete information from the server. The server can omit information about the environment cells, but it will not provide any incorrect information. The actions of the agents can also fail.

The simulation server provides the following perception information to each agent:

- Absolute position of the agent in the grid
- Contents of the cells that are within the agent's viewing range. If two agents are in each other's viewing range, they will be able to recognize whether they belong to the same team or not.

Agents are able to perform one action in each step. These actions are Skip, and movement in one of the eight directions (north, east, north east, ...). All actions except the Skip action can fail. An action failure happens either because of information distortion or because the conditions for successful execution of that action are not met. A failed action is evaluated as a Skip action by the simulation server.

The following list clarifies some specific cases that might happen during the simulation experiments:

- In case of two agents trying to move into the same grid cell, only one of the two actions succeeds. Determination of which agent succeeds to move into the cell is random in this case.
- It is impossible for two agents to swap places.
- A cow or agent can enter a grid cell only if that cell has been empty in the previous step.
- There is no restrictions on agents entering corral of the opponent team.
- If two agents or cows try to enter the same cell, determining which one succeeds is left to chance.

6.2.2 Target Description

Cows tend to move away from agents and obstacles and move towards empty spaces. The cows also like to get close to other cows and form a herd, but not too close. The herds tend to be tighter when the agents are close to them. Cows have a visibility range and an intimacy range which are both squares around the cow with the cow in the middle of the square. Cows get attracted to cows which are in their visibility range, and are repelled by cows that are in their intimacy range.

The cows are slower than agents. Each cow moves every three steps. The simulation server makes sure that not all cows move in the same step. The cows can not distinguish between empty cells and corral cells.

6.3 Cooperation and Task Decomposition

Based on the described cow movements, it becomes essential to have more than one agent for accomplishing the task of herding. Thus, we adopt a strategy that is strongly tied to the notion of groups of agents. Using the MOISE+ notation, we developed different types of organizational structure which determine how groups are formed and how the members of each group cooperate.

The organizational models provide the required structure for a herding group and an exploration group. The group formations and interactions are distinct and are determined by the type of the organizational structure. We employ three main models for organization and reorganization. In a coalition model, the agents only have communication relations in all groups. In a team model, certain agents within each group have authority relations. In a dynamic organization, agent relations within groups are changed based on the current task in hand. A decentralized reorganization is employed using which agents carry out the reorganization process [27]. The agents in the organization can send messages to the OrgManager to cause changes in the Organizational Entity (OE). Further details on the team and coalition structures can be found in [39, 38].

Agent roles within each organizational model are shown in Figures 6.1, 6.2, and 6.3.

For each group to be well-formed, all the specified roles for that group should be satisfied with at least their minimum cardinality. The agents that form each group with the required minimum and allowed maximum cardinalities are shown in Tables 6.1 and 6.2. As an example, well-formation

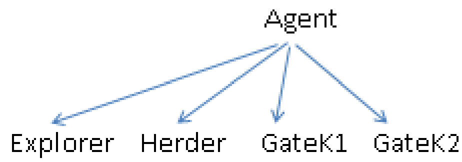


Figure 6.1: Agent roles used in a coalition organization

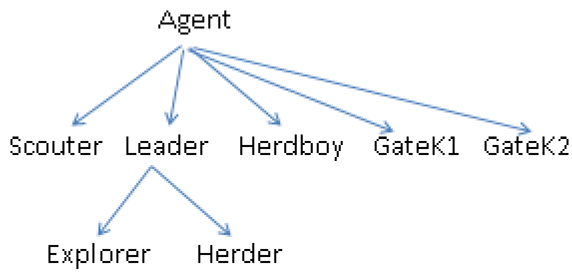


Figure 6.2: Agent roles used in a team organization

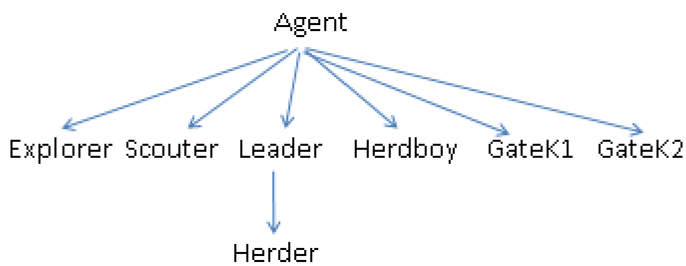


Figure 6.3: Agent roles used in a dynamic organization

of a herding group will require at least one herder for forming the group. At the same time, the herding group can contain up to ten herdboys and the two gate-keepers.

6.3.1 Agent Relations

Agents can have acquaintance, communication, or authority relations with each other. Each of these roles, preserves the properties of the previous role in the list. These relations are specified using the structural specification of each organization. For each of the organizations employed in this study, these relations are listed in Tables 6.3, 6.4, and 6.5. The Tables should be read as the relation of the role listed in the row to the role listed in the column. In addition to the relations specified in these Tables, there is an acquaintance relation between Gate keepers and all the other

Exploration group	Herding group
Explorer (1, 1)	Herder (1, 1)
Scouter (0, 1)	Herdboy (0, 10)
Gate keeper1 (0, 1)	Gate keeper1 (0, 1)
Gate keeper2 (0, 1)	Gate keeper2 (0, 1)

Table 6.1: Group formations in a team and a dynamic organization model

Exploration group	Herding group
Explorer (1, 2)	Herder (1, 10)
Gate keeper1 (0, 1)	Gate keeper1 (0, 1)
Gate keeper2 (0, 1)	Gate keeper2 (0, 1)

Table 6.2: Group formations in a coalition organization model

agents regardless of the organization model. There is also a communication relation between the two gate keepers in all organizational models.

6.3.2 System Dynamics

The general dynamics of the agents works as follows when the simulation starts. When there are no cows detected yet, the only groups that get formed are exploration groups. Once cows are detected, herding groups are formed to herd them into corrals.

When exploring, agents try to spread themselves out to cover a wider range and increase their chance of finding cows. Once a cow is detected, the agents try to form an herding group to herd a cluster of cows. The scenario requires two agents which need to cooperate and open the fence to allow cows and their team members to pass. Both the exploring and herding groups can be using

	Explorer	Scouter	Herder	Herdboy
Explorer	Communication	Authentication	Communication	Communication
Scouter	Communication	Acquaintance	Communication	Communication
Herder	Acquaintance	Acquaintance	Acquaintance	Authentication
Herdboy	Acquaintance	Acquaintance	Communication	Communication

Table 6.3: Agent relations in a dynamic organization

	Explorer	Scouter	Herder	Herdboy
Explorer	Acquaintance	Authentication	Acquaintance	Acquaintance
Scouter	Communication	Acquaintance	Acquaintance	Acquaintance
Herder	Acquaintance	Acquaintance	Acquaintance	Authentication
Herdboy	Acquaintance	Acquaintance	Communication	Communication

Table 6.4: Agent relations in a team

	Explorer	Herder
Explorer	Communication	Communication
Herder	Communication	Communication

Table 6.5: Agent relations in a coalition

the gate keeper agents to pass through the fence. Once either group reaches a closed fence, the agent with the gate keeper1 role is sent to the position of the first switch on the side of the fence that the group is. The gate keeper 1 activates that switch. This allows the agent playing the gate keeper 2 role to pass through the gate and go to the other side of the fence and position itself where the second switch gets activated. Before passing the gate, gate keeper 2 checks with all groups to see if any other group is trying to pass that fence. If that is the case, gate keeper 2 will communicate this to gate keeper 1 so the gate is kept open until all groups have passed the fence.

6.4 Organization and Reorganization Models using J-MOISE+

We consider organization as an already existing and predefined structure that the agents need to conform to. We found J-MOISE+ suitable for designing our problem solving agent organization. Similar to S-MOISE+, J-MOISE+ [29] is an open source organizational middleware that follows the MOISE+ specification. The overall system concepts between these two APIs are the same, and the main difference between the two is the supported language for programming the agents. In S-MOISE+, the agents are programmed using Java, while in J-MOISE+ the agents are programmed using Jason [5] which is an interpreter for an extended version of AgentSpeak [48]. The J-MOISE+

framework enables declaration of the organizational structure (role, groups, links), functioning (global goals, global plans, missions), obligations, and permission. Using J-MOISE+, the agents can be designed to receive an event when the status of a goal changes or when a goal is achieved.

Based on the roles used within each organization, the following organizational goals as demonstrated in Table 6.6 are defined and assigned to the roles.

The agents are also able to reason about their organization. We utilized the following main classes from the J-MOISE+ API:

- OrgBox API is used by the agents to access the organisational layer. OrgBox class has methods to generate organizational events like role adoption, mission commitment, group creation, etc.

- OrgManager agent is used to keep the current state of the organization and maintain it consistency. The OrgManager is able to receive messages from the agents OrgBox asking for changes in the organization's state (e.g., role adoption, group creation, mission commitment).

- OEAgent class is used to represent the agent inside the organization, as it stores the agents roles, missions, etc.

Figure 6.4 presents how the organizational layer gets incorporated in between the agent cominication layer and the general application layer. The agents access the organizational layer through the OrgBox API. The agents are written in JASON. KQML is used as the Agent Communication Language.

Using J-MOISE+, we can make sure that the agents within the system follow the organizational constraints. Using the J-MOISE+ framework, an Organizational Specification (OS) is defined which contains the static description of the organization. This description covers all different aspects of an organizational model such as types of roles and the behavior that an agent is responsible for once it adopts a role, interconnection between roles, groups, collective goals, how the goals

Role	Goal	Goal description
Explorer	findScouter	Find a free agent nearby to play scouter and help in the exploration
	findExplorer	find a free agent nearby to play explorer and help in the exploration
	shareSeenCows	share information about seen cows with other explorers in the group
	changeToHerding	Check if it is best to change to a herding group
	goToNearUnvisited	go to the nearest unvisited location (i.e., keep exploring)
Scouter	shareSeenCows	Share informarion about seen cows with other agents in the group
	followLeader	follow the leader of the group
Herder	recruit	recruit more herdboys depending on the size of the cluster
	releaseHerdBoys	release herdboys if there is too many of them
	defineFormation	calculate the ideal location of each group member
	moveToLocation	go to the location allocated to the agent in the formation
	changeToExploring	check if it is best to change to an exploring group
HerdBoy	shareSeenCows	share information about seen cows with other group members
	moveToLocation	move to the location allocated to the agent in the formation
GateKeeper1	gotoSwitch1(x,y)	move to a position to activate switch 1 at location x,y
	waitForGateKeeper2	keep switch 1 activated until GateKeeper2 has reached its location
	passFence	once the GateKeeper2 has reached its location, pass the fence and join other agents
GateKeeper2	gotoSwitch2(x,y)	move to a position to activate switch 2 at position x,y
	waitForAllToPass	wait until all agents in any group who wanted to pass the gate, have passed

Table 6.6: Organizational goals for each role in cow herding scenario

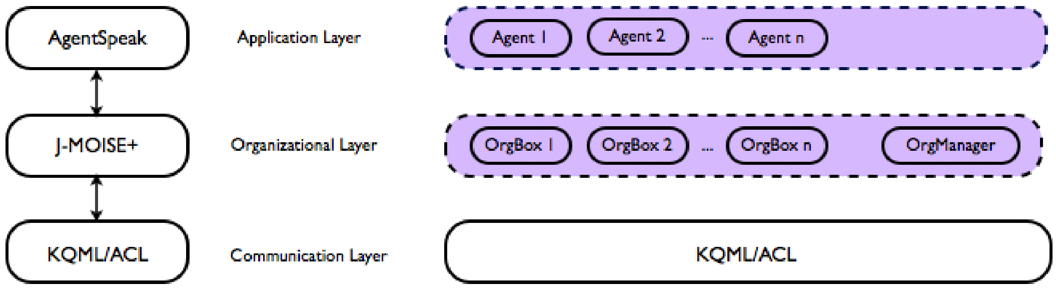


Figure 6.4: J-MOISE+ Organizational layer in relation to Application and ACL Layers

are decomposed into plans and distributed between agents, etc. The current instance of an OS is referred to as an Organizational Entity (OE). An OE is formed by a set of agents adopting an OS and thus having a common goal. J-MOISE+ employs an OrgManager agent that has the current state of the organization and keeps it consistent. Each agent in the system uses an OrgBox API to access the organizational layer.

Using the J-MOISE+ framework we have implemented coalition and team as two different models of organizational structure. We have also developed a task-based reorganization model which we refer to as Dynamic organization. We consider reorganization as a change in the current Organizational Entity (OE) which makes it possible to have a structural change between coalition and team. A short description of each of these models and the agent interactions using these models follows.

The structural specification created for each organizational model and reorganization, is presented in Figures 6.5, 6.6, and 6.7. These specifications map to the roles presented in section 6.3.

As stated previously, the agents are BDI agents written in JASON which is an interpreter for an extended version of AgentSpeak. AgentSpeak supports developing more complex agents than usual with a typical agent-based simulation toolkit. In particular, the language facilitates the development of agents with explicit representation of mental attitudes such as beliefs, goals, know-how

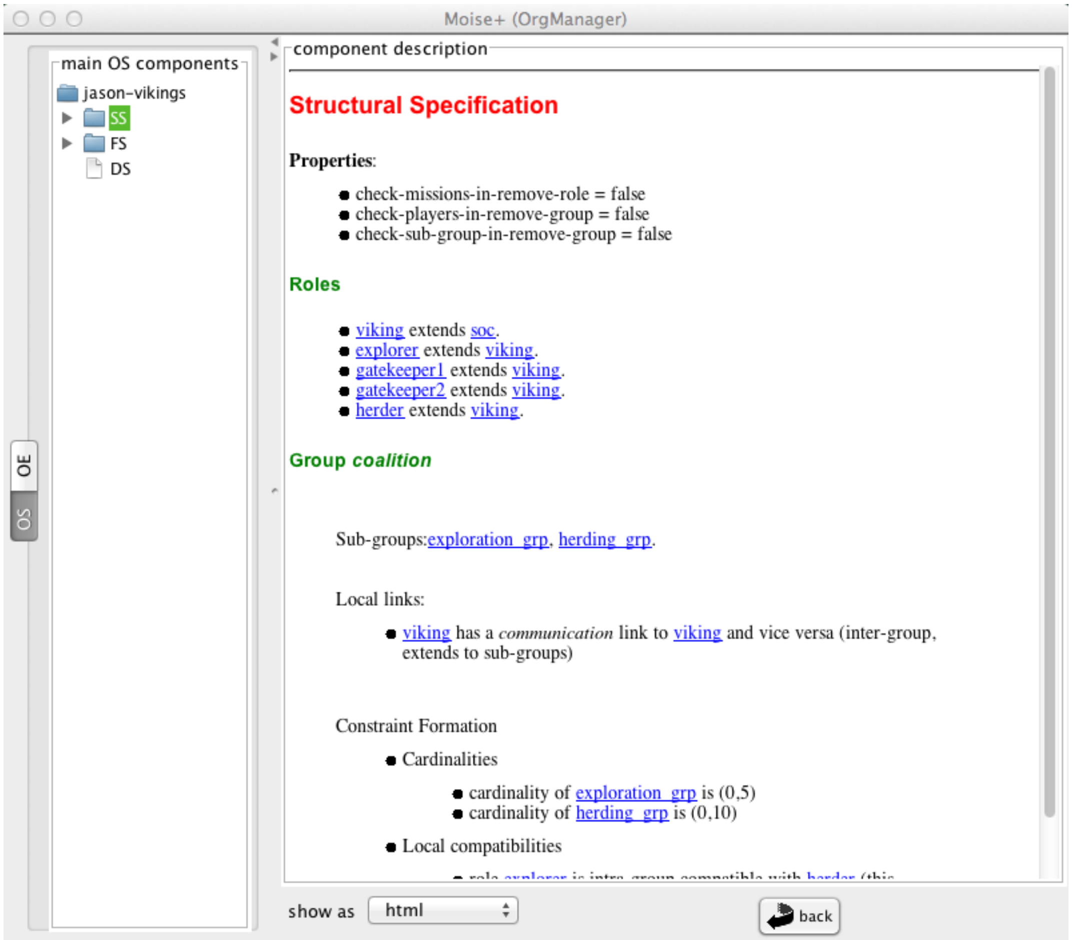


Figure 6.5: Structural Specification for coalition using J-MOISE+

(i.e., plans), and intentions. The notions of BDI are used to accommodate the following functionalities:

- *Beliefs* - Satisfies the informative aspects of each agent, i.e., the characteristics of the environment, specifications of the agent itself, and specifications of other agents of which this agent is aware.

- *Desire* - Satisfies the motivational aspects of an agent role, i.e., the objectives that should be accomplished, priorities of the objectives, and the payoffs associated to objectives.

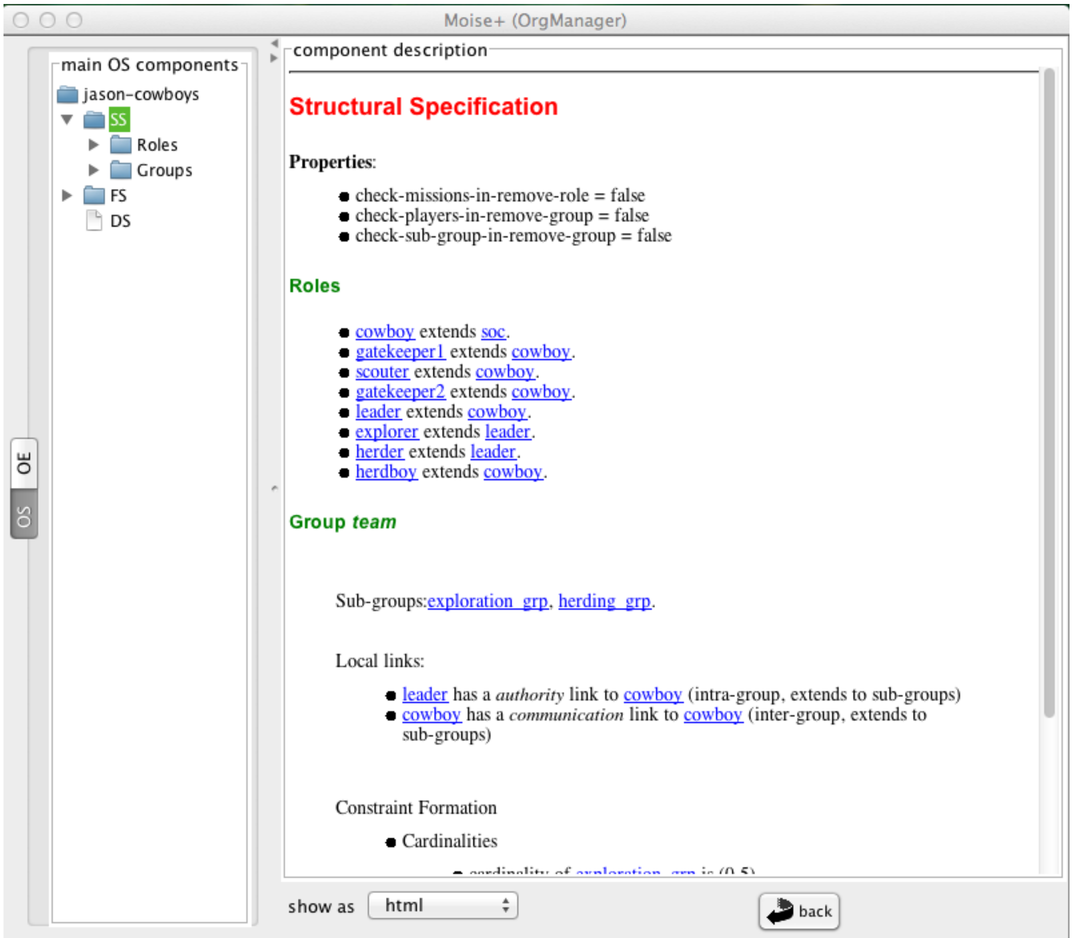


Figure 6.6: Structural Specification for team using J-MOISE+

- *Intention* - Satisfies a deliberative aspect of an agent role, i.e., the currently chosen course of action.

6.5 Experimental Setup

Different participating teams in the tournament correspond to the various organizational structures which are applied to the simulation model. Every simulation run is a competition between two teams with respect to a certain configuration of the environment. The winner of each simulation run is decided on based on the absolute number of cows which are caught. Both the simulation server and the participating teams are run locally on a 12-core system. Each team competes against

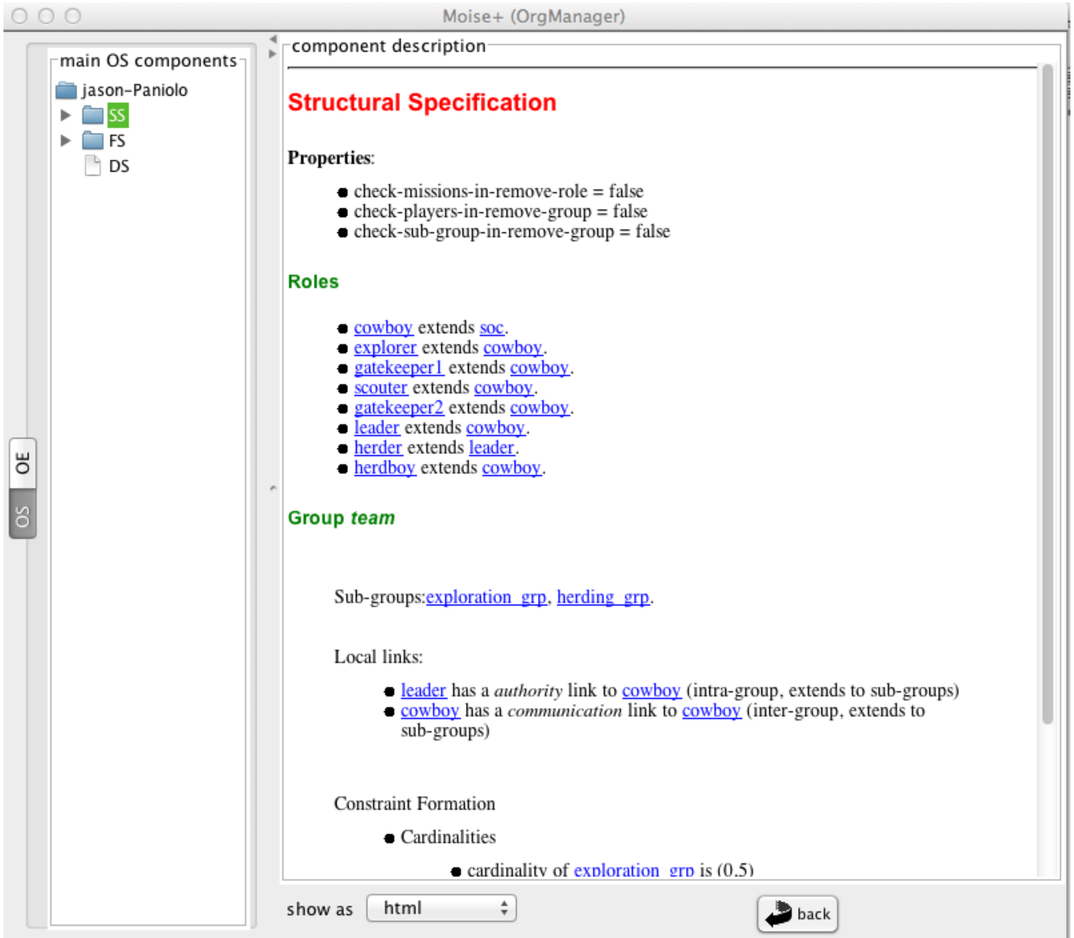


Figure 6.7: Structural Specification for dynamic organization using J-MOISE+

all the other teams in a series of matches. The winners from each match will then play against each other.

The agents from competing teams connect to the simulation server, identify and authenticate themselves before the match begins. Upon initiation, the agents receive information about the environment such as size of the grid, corral position, number of steps that the simulation will perform, etc. Each simulation consists of a certain number of steps. In each step, the simulation server (1) sends a sensory information to the agent(s), (2) waits for the agent reactions, and (3) processes the responses and calculates the next state of the environment.

The simulation server provides sensory information about the environment to the participating agents in a cyclic fashion, and expects their reactions within a certain time limit. The agents react to the received sensory information by indicating an action to perform in the environment. Lack of a reaction is interpreted as a Skip action by the server. The simulation server is stopped at 1000 cycles and notifies the participating agents about the end of a simulation.

Various test cases are designed to reveal organization performance and pose a range of movement complexity. We use three different levels of complexity provided by the simulation server. These levels of complexity are represented as different world models and are described. Snapshots of the graphical presentation of these different models are presented in Figures 6.8, 6.9, and 6.10.

Figure 6.8: 100 by 100 grid environment - World model 1

Table 6.7 lists number of objects in each world model. All the environments get populated with 20 agents, 10 agents per team.

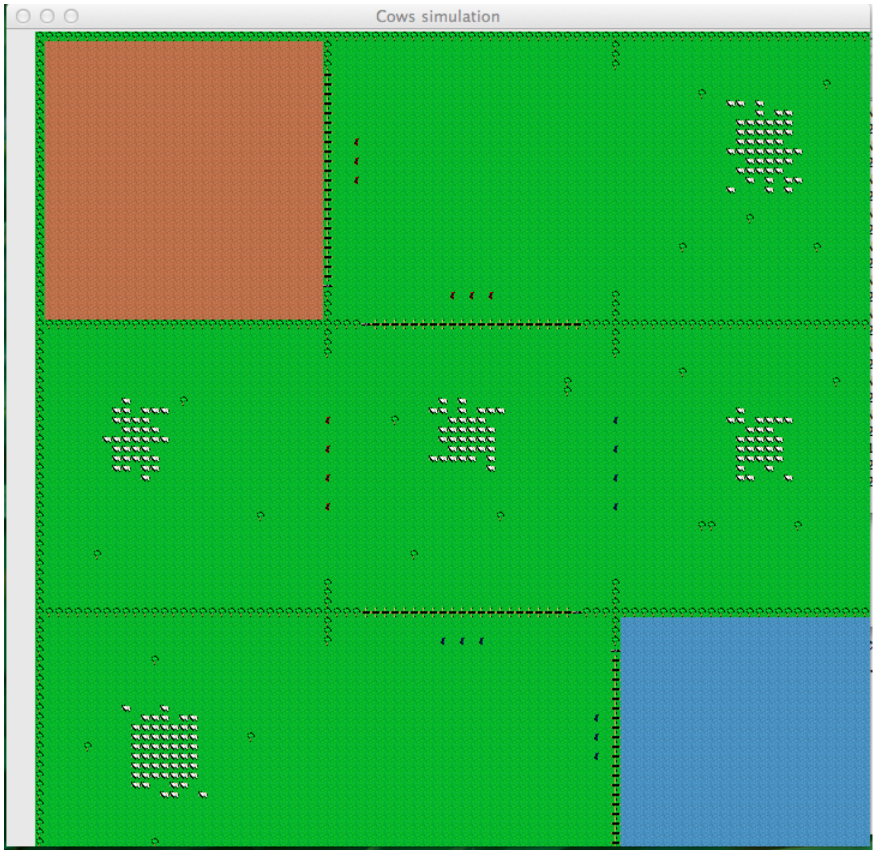


Figure 6.9: 90 by 90 grid environment - World model 2

- World model1 - There are no fences or gates in this model, but some obstacles. The model also has the least number of cows. The cows are widely spread out and the agents are also spread out between the cows.
- World model 2 - This model has more obstacles than world model 1. It also has fences and gates that need to be opened. The agents in this model are spread in three groups, from

	Grid size	Cows	Obstacles	Fences	Agent sight	cow sight
World model 1	100 by 100	131	163	0	8	5
World model 2	90 by 90	205	544	4	8	5
World model 3	80 by 80	406	692	4	8	5

Table 6.7: Variables in each world model

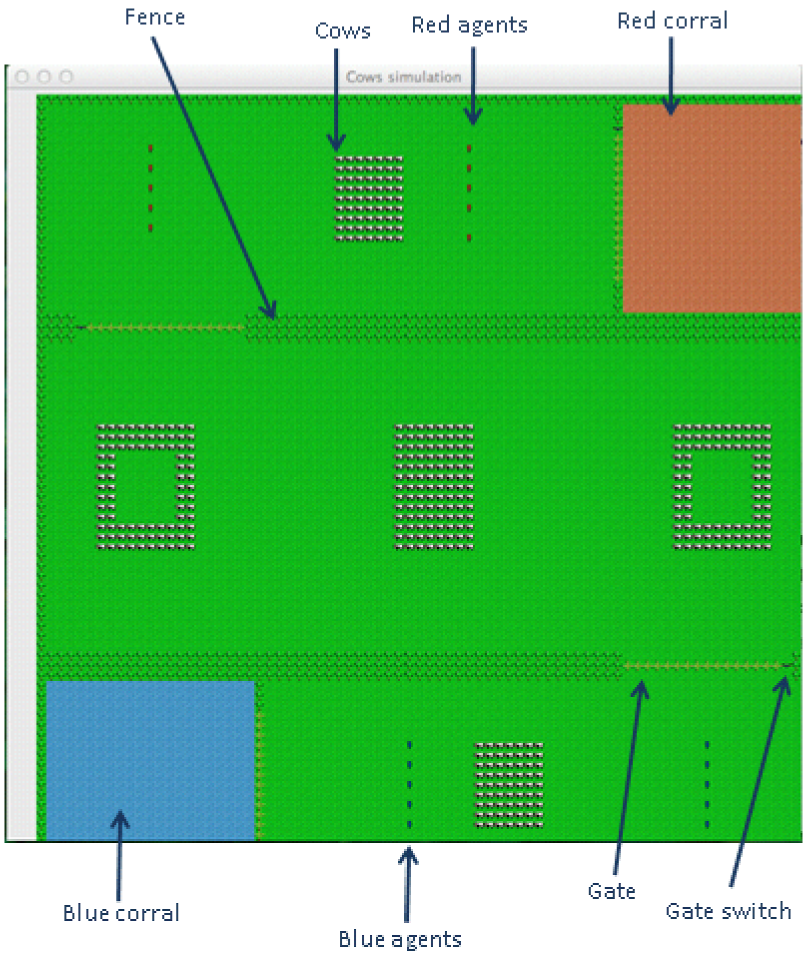


Figure 6.10: 80 by 80 grid environment - World model 3

which one group is in the middle of the world. The other two groups are closer to the team's corrals. The cows in this model are spread into five existing herds.

- World model 3 - This model has the highest number of obstacles. The model has the same number of fences and gates as World model 2. The agents in this model are the least spread out. They are situated as two groups in a close proximity to their own corral. This model also has the most number of cows. The cows in this model are spread into five herds.

	Team vs Coalition	Coalition vs Dynamic	Team vs Dynamic
World model 1	14:6	3:17	10:9
World model 2	9:11	1:19	9:11
World model 3	5:14	4:14	8:10

Table 6.8: Match results with actual number of wins for each team

	Team vs Coalition	Coalition vs Dynamic	Team vs Dynamic
World model 1	1 : 0	0 : 1	1 : 0
World model 2	0 : 1	0 : 1	0 : 1
World model 3	0 : 1	0 : 1	0 : 1

Table 6.9: Match results

6.6 Experiment Results

We conducted 20 simulation runs between every two teams. That is, matches between Coalition and Team structures, Coalition and Dynamic structures, Team and Dynamic structures. Table 6.8 shows a summary of win points for each competition. Performance is evaluated based on match results. The team who wins more matches is considered to have a higher performance.

The overall match results are presented in Table 6.9. In this Table, 0 represents a loss and a 1 represents a win.

6.7 Analysis

In matches between Coalition and Dynamic organization, Dynamic organization has a significantly higher number of wins. Which shows that there is a substantial difference between the two organizations.

In competition between Team and Coalition, the Coalition performs better on the two least spread world models, while the Team has a better performance on the most spread world model. This can be because the less spread models need a lot more exploration to find cows. In the more spread models, the task of exploration becomes easier, so there is not as much need for a Coalition structure and thus Team performs better in long run.

In summary, the dynamic reorganization gains a higher performance with the more complex world models. This proves our proposed methodology successful in improving the overall performance of organizational agents.

Looking into the results gained in between the Team and Dynamic models. The use of Coalition structure in the less spread models, helps the whole group of agents to gain a better overall result. While when the model is more spread, utilization of Coalition in the Dynamic model and reorganizing the agents affects the overall result in a negative way.

In comparison of the results between the Coalition and Dynamic models, the Dynamic model outperforms the Coalition in all world models. This is an expected result and re-emphasizes that different organizational structures applied to the same application will result in different performance characteristics. As in the more spread world model, existence of coalition is not a necessity for faster exploration, but it is still a preferred structure compared to a Team model. The Dynamic model enables the system to change to a Team structure when necessary and thus obtains a better result.

In summary, we find out that the dynamic reorganization provides a better performance when task complexity increases, thus proving our hypothesis. While in less complex environments and tasks, this reorganization can be less beneficial.

Chapter 7

Conclusion

This chapter provides a summary of the contributions of this work. We point out the lessons learned and the limitations of the system. At the end of this chapter we elaborate on the future work.

7.1 Summary

The primary objective of this work was to evaluate the potential benefits and feasibility of a strategic reorganization model that is based on inspirations from Social Organization Theory. Chapter 2 provided a background on Social Organization Theory and the concept of organization and re-organization in social systems. Several organizational structures are elaborated on and different organizational types are discussed. To further motivate the problem, in chapter 3 we discussed the problem of coordination and cooperation in multi-agent systems and how organizational design can be an effective means for handling agent coordination. Looking into the related work on applications of an explicit organization to multi-agent systems, we point out how in most related works, a single organizational structure is used throughout the system's lifecycle, despite the potential benefits that a system could gain from reorganizing to a different type of organizational structure. Thus, we look into the Intelligent Human Organizations framework and employ a multi-level control mechanism for enforcing organizational change.

To validate and evaluate the effectiveness of the suggested approach in improving overall system performance and to gain an understanding of potential limitations or constraint, in chapters 5 and 6 we demonstrated two case studies in which our proposed approachl for reorganization is utilized. These case studies are samples of geographically dispersed cooperative distributed problem solving applications. The MOISE+ framework is utilized to incorporate organizational design and reorganization into these two applications. A set of experiment runs compare the performance of each application with use of different organizational structures and reorganization. The results confirm that using the stretegic structural reorganization model, both applications gain a higher performance in more complex task and environment setups.

7.2 Contributions

As part of this research, a number of contributions have been made to the state of the art.

1. **We have developed a multi-level organizational control structure that allows a system to benefit from reorganizing to a different organizational structure despite the costs associated with reorganization** [41]. As shown in section 3.2.5., a lot of research has been performed on reorganization, but most of it is limited to reorganizing within the same structural model. For example, changing the size of holons in a holarchy. As a result, most systems do not utilize the benefits that can be gained from applying different types of organizational structure to different tasks within the same problem. To demonstrate that it is possible and beneficial to do so, we implement and utilize a multi-level organizational control model which uses a strategic management and an operative management layer to exert control and also to make interactions between these two levels of organizational control possible.

2. **We have developed a strategic, task-based model for triggering reorganization which allows the system to stay ahead of organizational change and gain higher performance in more complex problem settings** [38]. This was demonstrated through developing and applying the task-based reorganization model to an enhanced model of Pursuit game which requires cooperation between reactive agents with different skills to reach the goal of capturing preys. Results of experiment runs have demonstrated that the task-based reorganization model acheives higher performance once the system complexity increases. This is also an implicit indication that in less complex systems, the costs associated to reorganization can reduce the overall performance instead of increasing it.

3. **We have demonstarted how different organizational structures applied to the same application can result in different performance results** [38, 40]. This was demonstrated by applying different types of organizational structure to two different cooperative distributed problem solving applications and running experimental evaluations on them. The results of both experimental runs on both applications show how a different organizational structure performs better or worse for the overall application. These case studies and experimental evaluations were elaborated on in Chapters 5 and 6.

4. **We have presented a relationship between task complexity and effectiveness of reorganization** [40]. This is demonstrated through the experiments and the results gained from the experimental setups. The results demonstrated how efficacy of reorganization improved with increase in task complexity.

7.3 Limitations

In this section we point out any limitations and assumptions that this work has been subject to.

1. Closed versus open organizations - Our case studies and experimental work has been performed on closed organizations. Many real world applications might be subject to an open world model in which agents appear and disappear.

2. Invariant agents - Our case studies and experimental work are also limited to invariant agents, which implies the assumption that agent properties and skills do not change over time. While in some applications agents might learn skills or change properties.

3. Single group membership - We make the assumption that each agent can be the member of one group at a time.

4. We consider the communication and move costs to be the same, while in real-world applications these costs could be different.

7.4 Future Work

In this section we expand on issues that are worthy of further attention and evaluation.

1. Looking into invariant features of organizations can enable us to generate patterns of behavior which can be anticipated and acted upon accordingly. These patterns can be used for generating any association between organizational patterns and certain task characteristics. In future work we aim to develop a framework for categorizing different goals and task type characteristics that make them suitable for certain organizational structures. This will be accompanied by a learning method that will allow the system to learn from past experiences. Future work will also include trying to gain insights about the reorganization cost tradeoffs and the threshold upon which performance improvements can be gained from reorganizing agents despite the overhead costs of reorganization.

2. Another stream of future work can focus on frequency of task type changes and its relation to the effectiveness of our proposed reorganization model.

3. This work can also be enhanced by experimenting with other types of organizational structures.

4. Organizational learning can be another area for improvement which can benefit the system by learning from experience and applying that knowledge in future decision makings.

Bibliography

[1] http://multiagentcontest.org/2010. 48, 65, 66

[2] G. Allison and P. Zelikow. *Essence of Decision: Explaining the Cuban Missile Crisis*. 2 edition. 12

[3] L. Barton and V. Allen. Methods for Coalition Formation in Adaptation-Based Social Networks. In *Proceedings of the 11th International Workshop on Cooperative Information Agents*, pages 285–297, Berlin, Heidelberg, 2007. Springer-Verlag. 38

[4] C. Bernon, V. Chevrier, V. Hilaire, and P. Marrow. Applications of Self-Organising Multi-Agent Systems: An Initial Framework for Comparison. *Informatica (Slovenia)*, 30(1):73–82, 2006. 41

[5] R. Bordini, M. Wooldridge, and J. Hubner. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley and Sons, 2007. 72

[6] C. Brooks and E. Durfee. Congregation Formation in Multiagent Systems. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):145–170, 2003. 24

[7] K. Carley. Adaptive Organizations and Emergent Forms. In *Proceedings of the 3rd International Conference on Multi Agent Systems*, pages 2–3, Washington, DC, USA, 1998. IEEE Computer Society. 41

[8] K. Carley and L. Gasser. Computational Organization Theory. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 299–330. MIT Press, Cambridge, MA, USA, 1999. 2, 24, 25

[9] T. Carron and O. Boissier. Towards a Temporal Organizational Structure Language for Dynamic Multi-agent Systems. In *Proceedings of the 10th European Workshop on Multi-agent Systems: Modeling Autonomous Agents in a Multi-Agent World*, Annecy, France, 2001. 35

[10] V. Dignum and F. Dignum. ”structures for agent organizations”. pages 215– 220, 2005. 23

[11] V. Dignum, J. Vazquez-Salceda, and F. Dignum. A Model of Almost Everything: Norms, Structure and Ontologies in Agent Organizations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1498–1499, Washington, DC, USA, 2004. IEEE Computer Society. 27, 31, 36

[12] V. Dignum, J. Vazquez-Salceda, and F. Dignum. On the Engineering of Multi Agent Organizations. In *Proceedings of the 12th International Workshop on Agent-Oriented Software Engineering*, pages 53–65, 2011. 25

[13] E. Durfee. Scaling Up Agent Coordination Strategies. *Computer*, 34(7):39–46, 2001. 20

[14] E. Durfee, V. Lesser, and D. Corkill. Cooperation Through Communication in a Distributed Problem-Solving Network. In *Cognition, Computing, and Cooperation*, pages 159–186. Ablex Publishing Corporation, Norwood, NJ, USA, 1990. 24

[15] E. Durfee and T. Montgomery. Coordination as Distributed Search in a Hierarchical Behavior Space. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 21, pages 1363–1378. 1991. 38

[16] M. Esteva, J. Padget, and C. Sierra. Formalizing a Language for Institutions and Norms. In *Revised Papers from the 8th International Workshop on Intelligent Agents VIII*, pages 348–366, London, UK, 2002. Springer-Verlag. 27, 28

[17] J. Ferber, O. Gutknecht, and F. Michel. From Agents to Organizations: An Organizational View of Multi-agent Systems. In *Proceedings of Agent-Oriented Software Engineering: 4th International Workshop*, pages 214–230. Springer-Verlag, 2003. 27

[18] C. W. Fontaine. How Organizational Structure Impacts Organizations. In *Proceedings of the first Annual Conference on Organizational Effectiveness*, 2006. 16

[19] M. Gaston and M. desJardins. Agent-Organized Networks for Dynamic Team Formation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 230–237, New York, NY, USA, 2005. ACM. 37

[20] M. Ghijsen, W. Jansweijer, and B. Wielinga. Towards a Framework for Agent Coordination and Reorganization, AgentCoRe. In *Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 1–14, Berlin, Heidelberg, 2008. Springer. 3, 37

[21] B. Horling. *Quantitative Organizational Modeling and Design for Multi-Agent Systems*. PhD thesis, Department of Computer Science, University of Massachusetts at Amherst, February 2006. 32, 33, 55, 56

[22] B. Horling and V. Lesser. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2005. 25

[23] B. Horling and V. Lesser. Quantitative Organizational Models for Large-Scale Agent Systems. In T. Ishida, L. Gasser, and H. Nakashima, editors, *Massively Multi-Agent Systems*, pages 121–135. Springer-Verlag, Berlin, Kyoto, Japan, 2005. 2, 24, 25, 27

[24] B. Horling and V. Lesser. Using Quantitative Models to Search for Appropriate Organizational Designs. *Autonomous Agents and Multi-Agent Systems*, 16(2):95–149, 2008. 2, 24, 31

[25] B. Horling, R. Mailler, and V. Lesser. A Case Study of Organizational Effects in a Distributed Sensor Network. In *Proceedings of the International Conference on Intelligent Agent Technology*, pages 51–57, Beijing, China, September 2004. 3, 24

[26] J. Hubner, J. Sichman, and O. Boissier. A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence*, pages 118–128. Springer, 2002. 27, 42

[27] J. Hubner, J. Sichman, and O. Boissier. Using the MOISE+ for a Cooperative Framework of MAS Reorganisation. In *The 17th Brazilian Symposium on Artificial Intelligence*, pages 506–515, Berlin, 2004. Springer. 2, 3, 36, 69

[28] J. F. Hubner, J. S. Sichman, and O. Boissier. S-moise+: A middleware for developing organised multi-agent systems. *COIN I*, 3913:64–78, 2006. 30, 54

[29] J. F. Hubner, J. S. Sichman, and O. Boissier. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007. 72

[30] D. Isern, D. Sánchez, and A. Moreno. Organizational structures supported by agent-oriented methodologies. *Systems and Software*, 84(2):169–184, 2011. 27

[31] T. Ishida, L. Gasser, and M. Yokoo. Organization Self-Design of Distributed Production Systems. *IEEE Transactions on Knowledge and Data Engineering*, 4(2):123–134, 1992. 35

[32] R. Kota, N. Gibbins, and N. Jennings. Self-Organising Agent Organisations. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 797–804, Richland, SC, 2009. 2, 3, 37

[33] V. Lesser. A Retrospective View of FA/C Distributed Problem Solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1347–1363, November 1991. 20

[34] V. Lesser. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 1(1):89–111, 1998. 1, 20

[35] V. Lesser and D. Corkill. Functionally Accurate, Cooperative Distributed Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):81–96, 1981. 20, 24

[36] V. Lesser and L. Erman. Distributed Interpretation: A Model and an Experiment. *IEEE Transactions on Computers, Special Issue on Distributed Processing*, 29(12):1144–1163, 1980. 20

[37] R. Levitt and M. Nissen. The Virtual Design Team (VDT): a Multi-agent Analysis Framework for Designing Project Organizations. In *Integration of Knowledge Intensive Multi-Agent Systems*, pages 115–120, 2003. 27

[38] M. Mahani and A. Agah. A case study of task-based reorganization in a pursuit game simulation. In *International Conference on Artificial Intelligence*, pages 29–32, 2011. 69, 86

[39] M. Mahani and A. Agah. Task-based reorganization in multi-agent systems - exploring performance characteristics. In *International Conference on Agents and Artificial Intelligence (2)*, pages 215–221, 2012. 69

[40] M. Mahani and A. Agah. Strategic reorganization in geographically dispersed multi-agent applications. In *International Conference on Agents and Artificial Intelligence*, 2013. Under review. 86

[41] M. Mahani and A. Agah. Strategic Reorganization in Multi-agent Systems: Inspired by Intelligent Human Organization. *International Journal of Cooperative Information Systems*, 2013. Under review. 85

[42] E. Matson and S. DeLoach. Capability in Organization Based Multi-agent Systems. In *Proceedings of the Intelligent and Computer Systems*, Ljubljana, Slovenia, October 2003. 24

[43] R.E. Miles and C. Snow. *"Organizational strategy, structure, and process"*. Stanford University Press, 1978. 18

[44] M. Modarres. Reorganization: Contingent effects of changes in the ceo and structural complexity. *Academy of Strategic Management Journal*, 9:95–109, 2010. 2

[45] G. Morvan, D. Dupont, J. B. Soyez, and R. Merzouki. Engineering hierarchical complex systems: an agent-based approach. the case of flexible manufacturing systems. *Studies in Computational Intelligence*, 402:49–60, 2012. 3

[46] G. Picard and M. Gleizes. An Agent Architecture to Design Self-Organizing Collectives: Principles and Application. *Adaptive Agents and Multi-Agents Systems*, pages 141–158, 2002. 35

[47] T. Rahwan, S. Ramchurn, V. Dang, and N. Jennings. Near-Optimal Anytime Coalition Structure Generation. In *20th International Joint Conference on Artificial Intelligence*, pages 2365–2371, 2006.

[48] A. Rao. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In *Proceedings of the 7th European workshop on Modelling Autonomous Agents in a Multi-agent World : Agents Breaking Away*, pages 42–55, Secaucus, NJ, USA, 1996. Springer-Verlag New York, Inc. 72

[49] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohm. Coalition Structure Generation with Worst Case Guarantees, 1999. 24

[50] M. Shwaninger. Intelligent Organizations: An Integrative Framework. *Systems Research and Behavioral Science*, 18(2):137–158, 2010. 4, 17, 40

[51] J. Sichman and R. Conte. Multi-Agent Dependence by Dependence Graphs. In *1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 483–490, Bologna, Italy, 2002. ACM Press.

[52] H. Simon. *Models of Bounded Rationality*, volume 3. The MIT Press, 1997. 41

[53] Herbert A. Simon. *Models of Man: Social and Rational*. John Wiley and Sons, New York, 1957. 11, 12

[54] M. Sims, D. Corkill, and V. Lesser. Automated Organization Design for Multi-agent Systems. *Autonomous Agents and Multi-Agent Systems*, 16(2):151–185, 2008. 3, 23, 27, 28

[55] V. K. Singh, S. Husaini, and A. Singh. Self-organizing agent coalitions in distributed multi-agent systems. *Computational Intelligence and Communication Networks, International Conference on*, 0:650–655, 2010. 38

[56] Huynh T. D. Braines D. Smart, P. R. and N. Shadbolt. Dynamic Networks and Distributed Problem-Solving. *Knowledge Systems for Coalition Operations*, 2010. 38

[57] Y. So and E. Durfee. Designing Organizations for Computational Agents. In *Simulating Organizations: Computational Models of Institutions and Groups*, pages 47–64. MIT Press, Cambridge, MA, USA, 1998. 24

[58] P. Stone and M. Veloso. Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork. In *Proceedings of the 5th International Workshop on Intelligent Agents, Agent Theories, Architectures, and Languages*, pages 293–308, London, UK, 1999. Springer-Verlag. 3

[59] K. Sycara. Multiagent Systems. *AI Magazine*, 19(2):79–92, 1998. 41

[60] M. Tambe, D. Pynadath, and N. Chauvat. Building Dynamic Agent Organizations in Cyberspace. *IEEE Internet Computing*, 4(2):65–73, 2000. 34

[61] L. Vazquez and F. Lopez. An Agent-Based Model for Hierarchical Organizations. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 194–211, Berlin, Heidelberg, 2007. Springer-Verlag. 31

[62] M. Wooldridge. Intelligent Agents. In G. Weiss, editor, *Multiagent Systems*. The MIT Press, 1999. 2

[63] C. Zhang, V. Lesser, and S. Abdallah. Self-Organization for Coordinating Decentralized Reinforcement Learning. In Kaminka, van der Hoek, Luck Lesprance, and Sen, editors, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 739–746, Toronto, 2010. 38