Adaptive Interaction Protocol for Multi-Agent-based Supply Network

Khaled Ghoneim

A thesis

in

The Department

of

The Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science in (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

April 2009

Your file Votre référence
ISBN: 978-0-494-63168-3
Our file Notre référence
ISBN: 978-0-494-63168-3

ABSTRACT

Adaptable Interaction Protocol for Multi-Agent-based Supply Network

Khaled Ghoneim

Supply network is a special form of organization, where two paradigms are combined: market interaction and hierarchical structure. Because it includes different interdependent entities that should work together, supply network requires efficient tools to support coordination and improve the quality of interaction between these entities.

In this thesis, we propose a multi-agent-based interaction protocol for supply network formation. The main characteristic of this protocol is its adaptability to agent behavior during supply network formation. Such an adaptability is needed to form the most suitable organization structure. The novelty of the proposed protocol is the unification of contracting, auction and negotiation in a three-steps solution. Contracting is a planning tool aiming to facilitate solving network formation problem through task decomposition. Auction enables participating agents to jointly search an agreement space and check the contract feasibility. Finally, negotiation is the last resort for agents to reach an agreement. It is applicable when agents fail to reach an agreement due to lack of knowledge about the existing constraints. In fact, negotiation is used to release constraints when no realizable solution is obtained under auction. The second contribution of this thesis is the consideration of supply network formation within a general framework of coordinated distributed problem solving.

The proposed protocol and framework are simulated through a multi-agent system prototype serving as a proof of concept. The simulation results show the effectiveness of our three-steps solution protocol in terms of network formation success, customer satisfaction and the total gained rewards of the whole network.

Acknowledgements

This thesis is made possible through the support of so many wonderful people. First of all, I would like to take this opportunity to acknowledge all those who helped me during this thesis work. I would like to convey my deepest gratitude to my thesis supervisor Dr. Jamal Bentahar for introducing me to the world of Multi-agent systems, his valuable suggestions, expert guidance, and constant encouragement, during the course of this thesis work, and his enduring patience in reviewing my thesis.

I am grateful for the opportunity to study at the Institute for Information Systems Engineering CIISE at Concordia University for providing an excellent academic environment where there is opportunity to cross-fertilizes research ideas, Thanks go to the readers of this thesis, for their thoughtful comments, and I would also like to thank all the faculty and staff for their assistance during my master's course-work.

I am very grateful to Mrs. Wei Wan for her precious help in the implementation of the model I am proposing. Additionally, My colleagues at multi-agent laboratory of Concordia, for discussions on Multi-agent systems, their comments and suggestions. Special thanks to Mrs. Diane Fereig for her philosophical debates and her contribution during editing my paper.

Finally, I would like to thank my family, especially my beloved wife for their enduring support, encouragement, and endless love, your name should be on the title page next to mine. This thesis is dedicated to all of them.

Khaled Ghoneim,

April 14, 2009

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

"...every individual necessarily labours to render the annual revenue of the society as great as he can. He generally, indeed, neither intends to promote the public interest, nor knows how much he is promoting it. By preferring the support of domestic to that of foreign industry, he intends only his own security; and by directing that industry in such a manner as its produce may be of the greatest value, he intends only his own gain, and he is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention. Nor is it always the worse for the society that it was no part of it. By pursuing his own interest he frequently promotes that of the society more effectually than when he really intends to promote it. I have never known much good done by those who affected to trade for the public good." Adam Smith [116].

The purpose of this chapter is to present the context for our research, clarify the scope of this thesis and state our motivation, research goals and objectives. In Section 1.1, we present the general background of supply networks by exploring some definitions and explaining some key characteristics of supply network and organization coordination. In Section 1.2, we present the scope of the thesis and describe our motivations. In Section 1.3, we state the research problems under consideration. In Section 1.4, our research goals are clarified. Section 1.5 describes out our proposed solution and contribution. Finally, we present the structure of the thesis in Section 1.6.

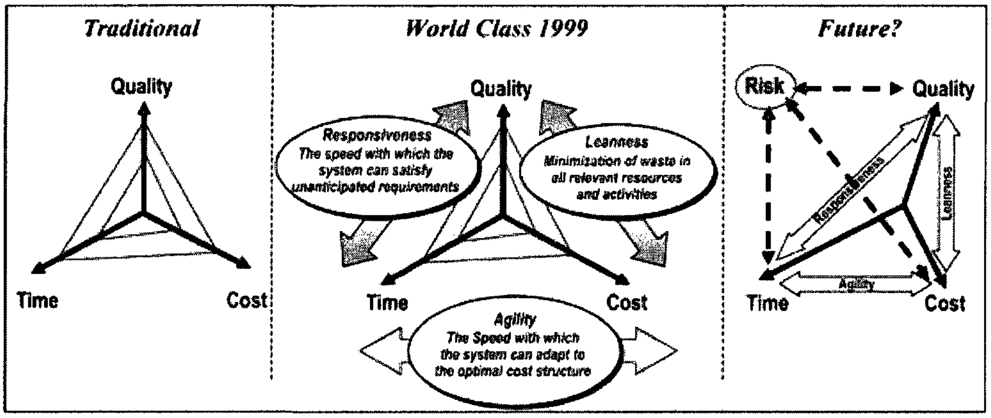
1.1 Background

1.1.1 Supply Network

Supply network (SN) is a network of autonomous business entities (i.e., customer, retailer, distributer, manufacturer, and supplier), and relationship between those entities through upstream and downstream links. In order to harmonize plans and activities and integrate shared resources and information, SN is considered as a (whole) system to capture business opportunities. SN may take any topological form and it does not have to be in a series configuration. As a whole system, SN goal requires execution of a precise set of actions by SN members (sub-systems). Those members are concerned with optimizing their own objectives that can be in conflict with each other. Consequently, it becomes quite important to find a compromise between these conflicting objectives.

SN is a "*life cycle process* comprising physical, information, financial, and knowledge flows whose purpose is to satisfy end-user requirements with products and services from multiple linked suppliers" [2]. SN is a " *network of organizations* that are involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services in the eyes of the ultimate consumer" [26]. Entities must perceive themselves not as self-contained competitive entities, but as nodes in a larger network of intersecting systems, composed of intricate, mutually supportive networks of customers, products, resources, enterprises, and information.

The tightening of the economy imposes leanness, and agility principles: A *Lean SN* "...employs continuous improvement efforts that focus on eliminating waste or non-value steps along the network; thereby achieving cost reduction, high capacity utilization rate, shorter lead times, and minimization of total supply network costs" [131]. An *Agile SN* "... profits by responding to rapidly changing, continually fragmenting global market by being dynamic and context specific. Achieving leanness and agility principles in SN reduces degree of freedom, which increases vulnerability to errors and uncertainty" [84] In order to be adaptive, SN achieves responsiveness to both planned and unexpected events through scalability of resource usage by mean of evolving its own organizational structure, (see Figure 1).



Source: Adapted from A. T. Kearney (1999)

Figure 1: Supply network concept development

Adaptive supply networks possess the flexibility to continually *morph* their organizational structure and respond to the environment in near real-time without compromising on operational and financial efficiencies. These networks seamlessly connect supply, planning, manufacturing, and distribution operations to critical enterprise applications and provide near real-time visibility across the supply network, thereby enabling rapid decision making and optimal execution [107]. This adaptability process requires an automated support for reconfigurable SN formation. A reconfigurable SN is a "rapidly configured multi-disciplinary network of small, process-specific firms configured to meet a window of opportunity to design and produce a specific product" [67].

SN changes its role in order to exploit business opportunities related to the *product life cycle*(PLC)[63]. PLC is getting shorter due to the competitive environment, and it becomes a cutting edge challenge. In order to exploit each PLC stage, we need a *dynamic supply* *network*, which is a temporary alliance of enterprises with integration and reconfiguration capability. The objective of dynamic supply network is to share skills and resources with the aim of better response to a business opportunity. When the opportunity is taken and the objectives behind forming the SN are achieved, the dynamic supply network is dissolved and a new alliance is made in order to achieve the necessary competitiveness to respond to another market opportunity [72].

Digital technological advances have propelled SN to an entirely different dimension by enabling globally information to be passed anywhere and any time between SN entities. This allows partners to create more channel oriented, competitive and optimized organizations. This spawned the engineering of *virtual supply network*, which refers to a socio-economic organization created to produce products/services and to make profit. For a virtual network to be reconfigurable, network entities should use intelligence and optimization techniques to solve problems and learning to adapt to a new environment, or to new knowledge. This leads to the *smart supply network*, which refers to an organization that is knowledge driven, inter-networked and dynamically adaptive to new organizational forms and practices to exploit business opportunities offered by the digital economy [36].

1.1.2 Supply Network Organization

SN as an organization is a social, goal-directed, coordinated and open system that must adapt to the environment. An effective organization structure allows organizational members to do the following:

- Achieve organization goals effectively.
- Deal with contingencies such as market changes.
- Gain a competitive advantage by developing the core enterprise and strategies to enable SN outperform others.
- Utilize resources efficiently in a cost effective way.

There are three organizational theories behind organizational structure design approaches [109]:

- According to classical organizational theory, effective organizations have a formal hierarchy, a clear set of rules, specialization and division of activities.
- According to neoclassical organizational theory, effective organizations are designed with flat hierarchical structures and a high degree of decentralization.
- According to the contingency approach, the best design for an organization depends on the nature of the environment in which the organization is operating.

Many organizations are shifting the paradigm from strict vertical hierarchies to flexible, decentralized structure that emphasize horizontal collaboration, widespread information sharing, and adaptability. *Chaos theory* [120] suggests that organization should be viewed more as a natural system rather than predictable machine. The ideas of chaos theory suggest that adaptive systems are nonlinear and made up of numerous interconnections and divergent choices that create unintended effects and render the universe unpredictable. However, chaos theory also recognizes that this randomness and disorder occurs within certain pattern of order. The global marketplace is another self-organizing chaos system. No one is in charge of the market, yet considerable coherence emerges from millions of independent, but connected, decisions.

SN organizational design refers to the process of coordinating the structural elements of organizations in the most appropriate manner. Essentially, SN organizational design is a process of configuration a space of parameters (tasks, roles, constraints, dependency, etc.), that controls the organization's behavior. Doing so requires addressing two primary issues: how to divide tasks, and how to coordinate the resulting tasks.

1.1.3 Supply Network Configuration

SN configuration problem is the assembly process of an organizational structure of participant entities given a local information, and inter-communication in order to transform resources into a composite goods/services. SN configuration demands both group coherence as well as competence. In essence, SN configuration formulates solutions by top-bottom problem decomposition into subproblems. Then allocates these subproblems, tackles each subproblem, exchanges subproblems solutions, and achieves a bottom-up assembling to synthesize these subproblems solutions into an overall solution.

The distributed problem solving approach for SN formation can be viewed as a composition of the following problems [32]:

- Goal specification: An SN has a *goal(s)*, which is to achieve a *task(s)* by a set of *activities* performed by *roles*, which are filled by *entities*.
- Problem recognition: The SN configuration begins by the recognition of potential out-sourcing, due to lack of ability, or of inefficiency to achieve the tasks.
- Goal decomposition: When faced with a complex task problem, agents solve it by reducing it into a set of smaller more manageable sub-problems. The goals and activities can be decomposed into sub-goals and sub-activities which are mapped to roles. The solution for these sub-problems can easily be determined. This approach deals with a problem in terms of roles that need to be played, and responsibilities associated with the roles. The central planning represents an *initial solution* (proposes a starting point for a solution) for preliminary centralized optimal plan for network organization structure, a centralized planning takes the form of *contract proposal* that specifies *roles*, and *skills* required to fulfill those preliminary distributed plans.
- Partners finding and selection: the formation of the SN organizational structure is supported by providing criteria to select the best partners which have the

matching skills to fulfill the required roles between competing entities.

- Goal alignment: A *Critique-solution* is provided through knowledge sharing in order to resolute conflicts by changing proposal criteria (change solution capacity with local requirement).
- Solution synthesis: A cycle of propose-critique-update is performed until a final plan is generated, and a social law (*contract*) that includes the norms, agreed upon contexts and contingency plans is established.
- Task(s) allocation: After decomposition, the final distributed plans are allocated to the entities that could potentially contribute to their implementation through commitments to the contracts.
- Emergence of network organization structure: Once the SN is formed, a relationship can be established between the agents, and each agent will perform its activity in order to achieve the assigned task(s).
- Monitoring the execution of plan: Adapting to changes in the environment through adaptive planning and application of social laws.

1.1.4 Supply Network Coordination

Coordination concerns creation of consistent distributed decision making network across the entire SN by aligning the self-interests of individual entities with SN's integrated interest. Coordination requires each node (entity) to take into account the impact its actions have on other nodes, hence enabling it to choose alternatives that optimize the SN's goal. The lack of coordination occurs either because different nodes of the SN have conflicting objectives, conflicting constraints, or incomplete information. Consequently, total SN gain is less than what could be achieved through coordination.

A coordination mechanism must be used to insure that every entity is informed of the SN changes and can make its own changes if necessary. Furthermore, the coordination between the SN entities necessitates well-structured plan. Planning is rational and structured decision making process, which aims to find the best choice of objectives and measures to a decision situation. Planning can be classified as *static* or *dynamic*. In *Static planning*, system designers and/or logistics planner select planning heuristics at the design time in order to choose what they believe to be the best decision for their specific application. Static planning provides high quality coordination, but tends to be rigid under dynamic system environment. Therefore, it proves to be incapable of taking account of unforeseen complex situations [54]. If during execution, one (or more) plan(s) for agents fails to progress as expected, the entire plan set is in danger of failing.

Adaptive planning is a continuous, event-driven and replanning process [77, 103]. In the adaptive planning, each entity monitors its plan execution through a repeated cycle of *plan-coordinate-execute* and if there is a new event that causes deviation, it stops all entities' progress, and triggers a partial or even full change of the previously accepted plan with long ripple effects. The development of adaptive planning is however a complex process. It requires knowledge-based tools, distributed decision making with parallel computation, learning and adaptation. *Contingency planning* is another means of dealing with dynamics, where each entity formulates not only its expected plan, but also alternative plans to respond to possible contingencies that can arise during execution time. These plans with their conditional branching are also merged and synchronized, although this imposes a significant computational and communication overheads, which can be saved by addressing plan deviation locally.

Planning can also be classified as *centralized and decentralized. Centralized planning for distributed plans* is a hierarchical command driven approach in which one entity acts as a central decisionmaker/ coordinator and the other entities act as a subordinates. A central decision maker/coordinator forms the optimal plans, analyzes them to find and eliminate any conflict or inconsistency between them, synchronizes entities activities then forwards them to its subordinates. Centralized planning is the conceptual framework underlying advanced planning systems software, which is offered by software vendors such as SAP™.

The main objective of this software is to consider the entire SN, obeying system constraints and accounting for interrelations between processes. Centralized computation gives better solutions, often faster, since communication is performed only twice (gather problem information and issue results). Centralized planning for distributed plans requires central coordinator to access all relevant information and to have the power to impose planning results on all organizational units. Empirical evidence implies that despite the benefits of centralized planning, the firms are reluctant to disclose private information to supply network partners. Therefore, this typically leads to incomplete information and sometimes infeasible plans. Another major shortcoming of centralized planning is the little support it provides with respect to negotiation process although it is the core of an effective collaboration.

Decentralized planning for centric plans is a planning process, which is distributed among numerous entities, each generates a partial plan. A central entity collects these individual plans together, then it analyzes the plans to discover what sequence of actions might lead to conflict, and modifies the plans to remove the conflict, then execute them similarly to centralized planning. The main advantage of decentralized planning is that it takes into account entities' constraints. *Decentralized planning for distributed plans* is a non-hierarchical approach to the coordination, where the planning process and its results are distributed. Decentralized planning for distributed plans approach partitions global complex planning problem into smaller specific problems, which are tackled by (local) independent entities through decomposition, aggregation and feedback mechanism. Planning usually achieved through communicative, negotiation-like process and grounds on consensus like agreements on objectives, measures and rules between partners. Decentralized planning for distributed plan serves to establish coordination between planning domains in order to create a common and mutually agreed upon plan [61]. This approach encompasses distributed problem solving, since it relies on entities being able to communicate about, tasks, solutions, goals, plans etc. On the downside, this approach is generally sub-optimal, since knowledge of the system is dispersed throughout the system and each entity has incomplete information about the system as a whole.

Coordination in SN cannot be accounted for without considering the social laws of the organizations and the way they constrain the behaviours of individual entities. Social law includes the rules, norms, agreed upon context, rewards of commitments and penalties of decommitments that aim to eliminate possible causes of conflicts. In addition, social law provides a system of ethics and social responsibilities.

To facilitate coordination, SN information resources ought to be organized and shared. The process of information integration in a SN can be decomposed into the following components: (1) standardization of information representation; (2) shared vocabulary; (3) problem-oriented data conceptualization; (4) transparency of information; and (5) standardization of information access and retrieval. By presenting information in a uniform self consistent way, it can be made sure that SN entities share the same view of SN, where the same information stored in more than one of the local systems should have the same meaning. Successful SN achievement requires understanding of both the customer requirements (i.e., quantity, lead time, product specification, components information, price, etc.) and SN functions information (SN echelons, service level, lead time capacity, process capacity, etc.). The relationships and properties among these concepts serve as the basis for the SN ontology. Ontology is a "formal explicit specification of shared conceptualization" [44]. In this regard, ontologies can be though of as formal semantic primitives that specify objects in a particular domain of knowledge. Ontology usually consists of a set of hierarchical classes or concepts, their definitions and axioms about them. It is necessary for an ontology to provide a sufficient level of details to describe the goods or services in their entirety in order to distinguish products from each other, as well as to describe attributes and features that are necessary to determining the value of the product/service. The main advantage for having a set of standard ontologies for the supply network is to facilitate the knowledge sharing re-use among various parties interested in that particular domain of knowledge. This will enhance the inter-operability between various SN entities. Organizational models such as SCOR and CPFR are basis for supply network ontology modeling since they are widely shared supply networks concepts [45, 48, 85, 137]. We discuss organizational models in further details in (Subsection 4.4.1).

1.2 Motivations

There are three distinguished decision levels depending on the time horizon: strategic, tactical and operational. As stated in [113], the strategic level deals with decisions that have a long-lasting effect on the enterprise. These include decisions regarding the number, location and capacities of warehouses and manufacturing plants, or the flow of material through the logistics network. The term supply network design is often employed as synonyms of strategic supply network planning [25, 75, 113]. To achieve this, an ideal network must have the optimum size and location.

Supply network is a special form of a network organization. It can be regarded as cross-hybrid between a pure-market chaotic interaction and a hierarchical organization relationship. It tries to combine the best features of the two. Entities behavior change throughout SN formation stages from competitive to cooperative.

Supply network formation refers to organizational design, which is the process of coordinating the structural elements of organization in the most appropriate manner. Design decision depends on specialization and division of task, authority and delegation, span of control, and information technology. SN emerging structure follows *contingency approach* where the best design for an organization depends on the nature of the environment in which the organization is operating. Therefore, SN organization is an organic rather than a mechanical process. As such it requires specific tools of interaction to support efficient planning and execution of the order fulfillment process. Supply network formation life cycle resembles a cooperative distributed problem solving process. The Supply network formation process takes place in a dynamic and often unpredictable environment. Supply network is considered as a system rather than a component, therefore, SN integration viewed as the process of seeking global rather than local optimization. This represents a great challenge, since each supply network has its own unique and complex features. In fact, supply network integration requires creating specific simulation models and solution techniques as a decision support system. To fulfill the modeling and simulation requirements of Supply network domain, multi-agent systems (MAS) technology is an appropriate candidate. This is because agents in MAS have varying capabilities and can interact and cooperate with each other. The primary economic theories of Adam Smith: the division of labor and the principle of laissez faire (free enterprise) [116] is the fundamental basis for our choice of multi agent system-based supply network. Adam Smith's "invisible hand" described the natural force that guides free market capitalism through competition for scarce resources, to trade in the most mutually beneficial manner. According to Adam Smith, in a free market, each participant will try to maximize self-interest. The interaction of market participants leads to exchange of goods and services, this enables each participant to be better off than when simply producing for himself/herself.

In the digital era, supply network planning should make full use of the technologies enablers such as web computing, distributed computing and process integration. Furthermore, these systems should be constructed by elements that are intelligent and autonomous. Agent computing and multi-agent systems technology can be regarded as an important approach in planning supply network formation. Multi-agent systems (MAS) are software systems where a set of intelligent autonomous agents interact with each other to perform a task. Agents are considered autonomous entities with social ability, reactivity and proactivity properties. Their behaviours can be either cooperative or self-interested. Multiple agents interact to coordinate their behavior in order to solve problems together.

In this domain, decision theory, as a mathematical theory of decision making, defines a *rational agent* as the one who chooses to perform actions that maximize *expected utility*. The problem with decision theory it is normative, in the sense that it tells us what agent should do in principle. On the other hand, it has nothing to say about how we might efficiently implement the desired behavior. In addition, in order to find a deal with optimal utility we need an unconstrained search over the space of all deals and their outcomes. Since any agent that we can implement on a computer must have resource bounds, such a search is prohibitively expensive particularly where an agent needs to consider a plan. Also, decision theory is quantitative in nature.

Allowing agents to use their autonomous and reasoning feature through negotiation will make them able to persuade each other. This will provide another mean of reaching agreement in a conflict of interest situation that was never attained before. As a result, the quality of reached deals will be improved.

Since multi-agent systems are systems in which multiple agents interact to solve problems, agent-based supply network formation process can be considered as a set of interactions between the supply network contractor(s) and supply network contractee(s). Key concepts of multi-agent systems are agents and agent coordination. Multiple agents will not be able to coordinate to achieve global goal without communication. Interaction protocols are "recipes" for structured communication between agents. An agent communication protocol consists of: (1) a set of messages encoded in an agent communication language (ACL), and (2) a set of rules governing the sequences of messages (conversation) [96].

The Foundation for Intelligent Physical Agents (FIPA) interaction protocols are well specified and widely used in MAS. However, they only describe the sequence of allowed actions between agents without any reasoning. Therefore, there is a need for a more flexible protocol, where agents have more selectable behavior. Currently existing MAS protocols such as *Contract Net* protocol[118] are used to do the initial planning, finding candidates, and distribute tasks. However, it does not find a solution when there is conflict of goals/constraints between entities. Furthermore, "Contract Net" do not offer argumentation in order to release the conflicting constraints, and to reach a win-win situation for all supply network participants.

1.3 Research Questions

We are motivated in answering the following research questions:

- *Which interaction mechanism approach is suitable as a general purpose interaction protocol for supply network formation?*
- *How does our interaction protocol limit the number of rounds, increase the number of reached agreements, and achieve better individual utility?*
- *How can interaction protocol adapt to dynamic agent behavior?*

Answering these questions is undoubtedly complex, given how difficult humans find it to rationalize the supply network problem solving and decision making process. Therefore, we do not expect a comprehensive answer to all these questions. However, we do intend to uncover issues the basic involved issues.

1.4 Research Goals

This thesis is concerned with the role of adaptive interaction protocol in supply network formation within the context of multi-agent system. Interaction protocol can bring several advantages to supply network such as adaptive reconfiguration [42]. Our aim, therefore is to find a new adaptive interaction protocol model that is selected by agent, in order to adapt to different phases of supply work formation process. In more detail, this thesis aims to:

- Study the environment of next generation supply network;
- Study and enhance the supply network configuration process;
- Study different coordination approaches;

- Study the issues and difficulties involved in supply network coordination;
- Analyze current approaches of supply network modeling to gain insight of available technologies;
- Propose an adaptive interaction protocol for supply network configuration;
- Provide a methodology for multi agent based supply network simulation;
- Design and develop multi agent based simulation tool for supply network configuration.

Building such artificial societies of autonomous software agents with suitable interaction mechanism for supply network configuration presents a very daunting challenge.

1.5 Proposed Solutions and Contributions

Our concern is to build an adaptive interaction protocol by taking the advantages of combining contracting, bidding and negotiation in order to improve the quality of supply network configuration. Our proposed interaction protocol is based on our analogy between supply network configuration and distributed problems solving. By providing a flexible interaction protocol that can adapt to changes in entities behavior and applying contracting, bidding, and negotiation in accordance with supply network configuration life cycle we hope to: (1) increase customer's satisfaction; (2) reduce the deal failure rate between agents; (3) improve the quality of solution by reaching a win-win situation.

A simulation model of supply network formation is also proposed using multi-agent technology with a case study about three-echelon supply network formation from three markets in order to:

- Show proof of concept in the feasibility and utility of intelligent agents in interaction on behalf of business entities in competitive and/or cooperative settings;
- Discover practical implications or limitations while designing such a system;
- Show that our proposed solution is feasible and useful.

1.6 Organization of the Thesis

The content of this thesis is organized into five chapters. Chapter 1 provides an over view of the thesis. In this chapter, we present the scope of the thesis and provide the specific motivations and objectives. Chapter 2 provides a clear definition and description of multi-agent organizational structure and coordination, and reviews the current literature of different multi-agent coordination approaches utilized in interaction protocols. This chapter also provides a critical analysis of these approaches with the intent of finding an appropriate solution with regards to the underlying difficulty. Chapter 3 presents the desiderata, and describes how the proposed solution could be implemented. In Chapter 4, we discuss a brief literature review of different modeling and simulation approaches. We also justify our choice of MAS-based modeling and simulation. We describe an example of such implementation by presenting a simulation prototype of a case study used to validate the interaction protocol we propose. Finally, we conclude in Chapter 5 by providing our contributions, limitations and our suggestions for future work.

Chapter 2

LITERATURE REVIEW

"It is not from the benevolence of the butcher, the brewer, or the baker that we expect our dinner, but from their regard to their own interest. We address ourselves, not to their humanity but to their self-love, and never talk to them of our own necessities but of their advantages." Adam Smith [116].

In chapter 1 we provided an overview of thesis, and this chapter explores the related issues and techniques in more details. In Section 2.1 we provide a brief introduction for multi-agent system theory, organizational structure, and coordination. In Section 2.2 we critically examine the literature concerning coordination approaches. Finally, in Section 2.3 we introduce the framework of our interaction protocol.

2.1 Introduction to Multi-Agent System

Multi-agent systems (MAS) are comprised of a number of heterogeneous agents that work independently and in a collaborative manner to solve problems in a decentralized environment. MAS concentrates on the development of dynamic organizational structures and problem solution strategies for a range of distributed knowledge-based problem-solving modules (agents). Multi agent systems consist of self-contained knowledge-based systems that are able to tackle dynamic distributed problems. Thus, agents should be able to interact in order to inter-operate and co-ordinate with each other, and to learn from one another and from the environment. An agent's social environment is a communication environment in which agents interact in a coordinated manner [20]. According to dependency theory, agents are engaging in interactions because of the incompatibility between their goals, resources and capabilities [21, 91, 112]. The general goal of MAS is to create systems that interconnect autonomous agents. Thus enables the system to function beyond the capabilities of any singular agent. This is because the limited knowledge, computing resources, and perspectives limit the capability of the individual agent [86].

Various definitions from different disciplines have been proposed for the term MAS and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics [46, 51]:

- Each agent has incomplete capabilities to solve the main problem;
- There is no global system control to direct each entity's action;
- Data is decentralized;
- Each entity can make some decision independently and share their knowledge through communications;
- Computation is asynchronous.

2.1.1 Intelligent Agent

An agent is a computer system that is situated in some environment, and that is capable of autonomous actions in this environment in order to meet its design objectives[136]. Skolicki and Arciszewski [114] defined intelligent agent as an autonomous system situated within an environment. It senses its environment, maintains some knowledge and learns upon obtaining new data. And it acts in pursuit of its own agenda to achieve its goals, possibly influencing the environment. Researchers have proposed several properties that could distinguish an intelligent agent [52, 62, 102, 114]:

- *Autonomy*: An agent encapsulates some state (which is not accessible to other agents), make decisions about what to do based on this state, without the direct intervention of other agents;
- *Perceiving*: Through sensors, an agent should sense and observe the environment within which it is located;
- *Pro-activity*: Agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy their design objectives;
- *Reactivity*: Agents are situated in an environment (which may be the Internet, or a collection of agents), where they are able to perceive this environment through the use of sensors, and respond in a timely fashion to changes that occur in it;
- *Sociability*: Agents are capable of interacting with other agents through a given communication language. Therefore, they are able to explore, discover, negotiate, and coordinate, and may be required to understand and reason about the goals of others;
- *Rationality*: Agents are economically *rational* in the sense of never accepting a deal that would not improve their individual welfare. Agents that follow the notion of individual rationality are called "myopic", because they only consider immediate gain. Any sequence of rational deals will eventually result in an allocation of tasks with maximal social welfare. This means (1) there can be no infinite sequence of deals all of which are rational; (2) once no more rational deals are possible, the agent society must have reached an allocation that has maximal social welfare [33, 73];
- *Ontology*: Agents need ontology to store and maintain knowledge to be able to understand each other;
- *Adaptability*: An agent needs to learn to enhance its behavior to adapt to environment changes.

In terms of agents' behavior attributes we distinguish between cooperative and non-cooperative agents; *Non-cooperative*: Agents are self-interested and each agent tries to maximize its own utility. In such as setting, there is no notion of global utility. Accordingly, agents play *competitive strategy*, where each agent has his own individual goal. In our SN formation problem, the main tools that could be used with agents having such attitude is bidding and negotiation. SN accounting systems reimburse competing agents on the basis of individual contribution rather than on overall organizational performance. *Cooperative*: Agents share the same utility function to build the global utility function. Their goal is to collectively maximize the global utility. Consequently, agent play cooperative, where agents has shared goal. To cooperate successfully, each agent must maintain a model of the other agents, and also develop a model of future interactions, where sociability is presupposed. Our work will focus on both cooperative and non-cooperative agent behavior and its effect on the overall system performance since agents change their roles during the interaction according to prevailing circumstances.

Agent Theory

Agents with *beliefs-desires-intentions* are known as a BDI agent. BDI is a well-known method to describe rational agents based on practical reasoning theory originally developed by Michael Bratman [11]. BDI architecture is based on Bratman's philosophical model. The BDI model addresses how beliefs, desires and intentions are represented, updated, and processed. BDI describes an agent's beliefs about the system and environment, the agent desires (or goals) to achieve as well as expressing the agent's intention by way of executable plans.

An agent's Beliefs corresponds to information the agent has about the current state of the environment, including themselves and other agents. Agent's beliefs are described as a set of sentences in a formal language together with deductive process for driving the consequences of those beliefs. It is the basis for all of its argumentation, reasoning, planning and subsequent actions. Beliefs can also include inference rules, allowing forward chaining to lead to new beliefs. In general, this information is stored in a database that is usually called a belief base, which may not be necessarily consistent.

Desires are things that agents would like to accomplish. They are the motivational state of the agent. An agent's desire represents states of affairs that the agent would, in an ideal world, wish to be brought about. Desires are the assignment of goodness to the state of the world, from the agent's perspective.

Intentions are the agents' targets. They represent the deliberative state of the agent: what the agent has chosen to do, the agent's committed plans, or the course of action to take. Plans are sequences of actions that an agent can perform to achieve one or more intentions. Bratman's theory focuses on role that *intentions* play in practical reasoning. He argues that intentions are important because they constraint the reasoning an agent is required to do in order to select an action to perform. This reduction of number of possibilities makes decision making much simpler than would otherwise be the case. An agent reasons about what is the best plan for achieving its desires under specific beliefs about the environment. An agent can review its goals and respond with revised plans, if necessary as environmental parameters change.

BDI model is attractive for several reasons [99]:

1. It is founded upon a well-known and highly respected theory of human rational action [11];

2. It is intuitive (what to do and then how to do it);

3. It gives a clear functional decomposition, which indicates what sorts of subsystems might be required to build an agent;

4. It has been implemented and successfully used in a number of complex fielded applications;

5. The theory has been rigorously formalized in a family of BDI logics;
6. BDI model is useful in situations of incomplete knowledge or limited computational resources.

Although mental primitives, such as beliefs, desires, and intentions, are appropriate for a number of applications, they are not enough by themselves for understanding all aspects of social interaction.

Agent Communication

In MAS, BDI-agents needs rich communication and domain language and well defined interaction protocols to be able to exchange meta-level information.

The communication language should provide enough and suitable locutions to allow agents to pass call for bids, proposals, counter-proposals, acceptances, and rejections as well as meta-information. Elements of communication language are usually referred to as *locutions* (a particular work, phrase or expression) or *utterance* (manner of speaking). The domain language should express proposal (e.g., describing products available for sale) as well as meta-information about the world, agent's beliefs, preferences, goals etc.

A multi-agent system consists of a group of agents that interact with each other. This interaction is generally regarded as the foundation for cooperative and competitive behaviours in autonomous agents. The term interaction protocol is used in reference to a set of rules that guide interactions. The interaction protocol specifies, at each stage of agent conversation, who is allowed to say what. An agent interaction protocol defines sequence of information, by mean of describing a communication pattern as an allowed sequence of messages between entities and the constraints on the contents of those messages. Interaction protocol may address the following issues: rules for admission, rules for participant withdrawal, termination rules, rules for proposal validity, rules for outcome determination, and commitment rules.

In order to achieve efficiency in SN domain, agents must have an efficient way to select right strategies, and the related interaction mechanism. SN formation task cannot be performed without an interaction method that enables allocating tasks, communicating information, aligning performance measure along nodes and maintaining interaction between nodes. To avoid chaotic behaviours of agents, a well-designed interaction mechanism is conducive to making MAS operate smoothly. Rosenchein and Zlotkin [101] propose five attributes necessary for a 'good' interaction mechanism: efficiency, stability, simplicity, distributivity and symmetry. This will be the basis of our critical discussion about coordination mechanism and an inspiration to our new proposed framework.

2.1.2 Multi-Agent Organizational Structure

In a dynamic environment, a MAS may encounter new situations where organizational structure is no longer the most effective, therefore the organizational design of MAS determines the performance of task achievement. Since SN organization is dynamic, SN should re-design itself to meet task environment changes. Agents within SN are reconfiguring themselves by learning from newly acquired knowledge. *Self-organization* means the process of generating, adapting and changing organizational structure, which is the result of individual choices by a set of agents to engage in interaction in certain organizational patterns. Consequently, MAS should be *self-building* (able to determine the most appropriate organizational structure for the system by themselves at run-time) and *adaptive* (able to change this structure as their environment changes). As a result, as organization changes, the process, communication, and types of interaction change.

It is the structural organization of agents that gives a MAS the knowledge, competence, ability and synergy provided by all its individual agents and represents the value-added with the multi-agent approach. Organizational structure can be used to control the complexity of both the design and configuration of MAS and of the execution by individual agents. In fact, it is possible to change the behavior of the MAS without changing the constituent agents merely by altering the organizational relationships between them. This means that the very essence of a MAS implies that organizational structure of the societies of agents has to be taken into account during analysis, design and implementation. Various models of self-organizing MAS-organizations have been built. For instance, Turner & Jennings [126] use self-organization for scalability issues in MAS, where organization plays an important role. They improve system performance by the individual agents ability to determine the most appropriate communication structure for the system by themselves at run-time and to change this structure as their environment changes.

Agent-based organizational structure design may be imposed or emerged spontaneously. When imposed, the organizational structure can be regarded as top-down design problem, where solution space is explored and candidate design is evaluated before its implementation [31, 90, 93]. Alternatively, it can be viewed as a bottom-up design problem, where boundaries between agents are determined as problem solving progresses, and the adjustments to the structure are made in response to current structure performance inefficiencies to current environment state [47].

The supply network configuration problem is the process of assembling complex solutions and the exchange relationship between autonomous self-interested agents in order to determine the participants in the SN. Solving the SN configuration problem requires collective effort. SN configuration resembles *coordinated distributed problem solving* using agents. These agents have specialized capabilities and can perform only certain combinations of tasks, or produce certain resources. In order to complete a complex task, an agent may delegate subtasks to other agents, which may in turn delegate further subtasks. Agents need to interact to coordinate their effort in such problems. In coordinated distributed problem solving, each agent is provided with an interaction protocol (possible actions that agents can take at different points of the interactions) and a strategy (a mapping from state history to action, a way to use the protocol) in order to interact effectively.

Agent-based organizational structure is founded upon problem decomposition structure, which defines roles, responsibilities, and preferences for the agents within a society, and this in turn defines control and communication patterns between them. This organizational intelligence comprises many dimensions, including communication capabilities, knowledge about who knows what and knowledge about norms and procedures. The organization of MAS consists of many aspects such as interaction patterns between agents, agents roles, and coordination mechanism [19, 49, 138]. A multi-agent based organizational structure has a set of oriented functional positions *agents' roles* to coordinate the activities of the agents within the organization for the production of goods and services. An organizational structure is represented as a set of rules (*interaction protocol*) for when to communicate what to whom, and how to structure the communication. These rules provide strategic-level mechanism for dynamically re-configuring organizational structure. Coordination mechanism is the main organizational aspect of adaptive MAS. Therefore, dynamic adaptation of coordination mechanism is the focus of this dissertation.

2.1.3 Multi-Agent Coordination

If a problem domain is complex, large, or dynamic (such as in supply network domains), then the only way to address the problem is by developing a number of agents that specialize in solving particular problems. This decomposition allows each agent to use its best knowledge for solving the particular problem. Thus, when and inter-dependent problem arise, the agents need to coordinate or collaborate with one another to ensure that the tasks' interdependence are properly managed from different perspectives.

Coordination can be defined as managing dependencies between activities [69] and it includes managing shared resources, task assignment, and task/sub-task dependencies. In multi agent systems, an agent often find benefits in coordinating with other agents, through gaining information or performing actions toward goals in order to achieve a payoff. Effective coordination is essential if autonomous agents are to achieve their goals in MAS. Such

		Buyer	
		One	Many
Vendor	One	Negotiation	Auction
	Many	Reverse Auction	Market

Table 1: Market mechanisms

coordination requires management of the various forms of *dependency* that naturally occur when agents have inter-linked objectives, when they share common environment, or when they share resources. Ferber [35] described coordination forms, they are mainly: planning, reactivity and regulation; Our main concern are:

- Coordination by planning: This technique includes two phases. In the first, we consider the set of actions to be carried out to achieve a goal and produce a set of plans. In the second, we select one of the plans, which we then execute. It may be necessary to change plans while they are being carried out, which therefore means that dynamic replanning options should exist. In MAS, the different plans drawn up by the agents may lead to conflicts of objectives, or conflicts over access to resources. Plans must then be coordinated in such a way as to solve these conflicts and so achieve the goals of the different agents.
- Coordination by regulation: The principle is to set rules of behaviours (contract terms) that aim to eliminate possible conflicts e.g., social laws
- Coordination by market mechanisms: Market mechanisms [79, 128] is a set of rules that governs interaction among buyers and sellers and determines how to form a deal in a fair and efficient manner, for different market mechanisms (see Table 1) [65].

Since actions performed by one agent constraints and are constrained by the actions of other agents, an agent cannot by itself just make a locally optimal decision, but must determine the effect its decisions will have on other agents and coordinate with others to choose and execute an alternative that is optimal over the entire supply network. The purpose of coordination is to allow agents to produce their local plans with minimum conflict with each other and the global plan. In summary, coordination is essential for following reasons [87]: (1) preventing chaos; (2) increase solution efficiency; (3) meeting global constraints; and (4) managing distributed information, expertise, and resources.

Agents benefit from coordination with other agents to solve an SN configuration problem. Coordination problems arise when: (1) there are alternative actions the agent can choose from, each choice affects the environment and results in different states of affairs; (2) the order and time of executing actions affects the environment, resulting in having agents in different states of affairs; (3) the local changes in the plan can impact other partners; and (4) an agent has incomplete knowledge of the environment and of the consequences of its actions and the environment changes dynamically making it more difficult for the agent to evaluate the current situation and the possible outcomes of its actions.

Coordination requires agents to identify how they could work together. The majority of the participating agents are neither obedient nor adversarial, but rational or self interested in the sense that each agent has its own goals and preferences. The decision for a self-interested agent to be cooperative is based upon whether the cooperation can bring greater benefit than working alone.

In order for agents to coordinate, they must simultaneously communicate about their plans at multiple levels so as to build a solution model of joint activities, and this requires an interaction mechanism.

2.2 Agent Coordination Approaches

In this section, we present a short literature review of basic agent coordination approaches to interaction protocol, and a critical analysis of each approach. Our goal is to provide the reader with general understanding of the different interaction approaches in order to find the most suitable approach. We argue that existing approaches have their own limitations, then we will show that negotiation based approach for conflict resolution has a potential to overcome these limitations.

2.2.1 Contracts

In the free market, SN participating members are not under sole control of a central authority. The problem is even more stringent due to incomplete information sharing. One way to improve the system performance is to achieve coordination among participating parties through establishment of contracts. When rational, autonomous agents encounter an opportunity to join some groups such as SN, they must decide whether to commit to doing the assigned task(s). To decide whether to join a proposed SN, an agent needs to asses (1) the impact of the commitment in terms of benefits, cost, and risk; (2) the possibilities for assigning tasks to other group members in an individually rational manner, the potential contribution the new members could make to group activity and the costs and risk of subletting the tasks.

Contract as a planning tool

Contracts represent coordination by planning. The purpose of the planning is to determine the future actions of agents.

A central decision maker or coordinator agent forms the optimal plans and analyzes them to find and eliminate any conflict or inconsistencies between them, synchronizes agents' activities according to schedule, then forwards it to subordinate agents [41] in a form of *contract proposal.* Since the availability of agents for the sub-plans is determined after devising the sub-plans, it is not certain that the decomposed plan can be allocated.

Agents formulate local plans for themselves as individuals. These plans may conflict with the central plan or the announced contract proposal distributed by coordinator agent. In order to solve this conflict, agents have to interact in order to generate consistent plans, by providing their own goals and constraints. The central agent coordinates those agents generated plans in order to generate a consistent synchronized *joint plan.* A decentralized planning approach encompasses distributed problem solving, since it partitions the global complex planning problem into smaller specific problems, which are tackled by agents through decomposition, aggregation, and feedback mechanism.

Contract as a social law

The main principle of social law is to set rules of behaviours that aim to eliminate possible conflicts. Coordination in organizations and societies cannot be accounted for without considering the social laws of the organizations and the way they constrain the behaviours of individual agents. A social law [111] is established to prohibit the agent against particular choices of actions in particular context. The participants enter into binding agreement with each other by agreeing on functional and quality metrics of the resources they request and provide. They commit to the contract and attempt to fulfill what they promised by achieving their intended goals. A contract represents an obligations and rewards for the participants in it.

By definition, a commitment specifies a pledge to do a certain course of action [49]. Commitments are binary relationships (legal abstraction)that binds/associates pair of agents. An *obligation*(delivered product, deadlines etc.) is a *norm* for which unfulfillment is always penalized. *Compliance* of the norm benefits its addressee agent. Therefore, contract can be represented with a set of norms that specify what must be done to consider a task fulfilled. Monitoring the performance of task execution based on the *norm* is an essential part of the process. Social commitments [50] are a special type of norms constraining the behavior of autonomous agents and the fulfillments of commitments which are the keys to successful coordination. Since commitments reveal agents' intention, it provides a degree of predictability so that agents can take the future activities of other agents into consideration when dealing with intra-agent dependencies, global constraints, or resource utilization conflicts. Communication is very important for the agent to decide what commitment to make since it provides the information it needs.

Contract as coordination tool

All the trade-off among SN partners are fulfilled by contracts, so all the means that help to achieve coordination will be restricted by contracts. SN contract is a coordination mechanism that provides incentives to all members so that the decentralized SN behaves as one unit. An SN contract terms regulate how partners do business together and are rewarded. Contract tools such as discount, penalty, and revenue sharing are a set of clauses that offers an incentives mechanism to guarantee all the entities in the SN will coordinate and optimize the channel performance. Other tools such as norms, decommitment penalty are used to control agent behavior during execution phase. SN coordination models using bilateral contractsusually deal with two independent parties. Each party's decisions in the SN are motivated by the local incentive of individual partners in the form of benefits, costs and the risk with respect to demand, supply or internal process. An SN contract specifies parameters governing the bi-lateral relationship, these parameters/contracts-termscan be used to entice individual parties to act in a globally desired way.

Contract as a portable ontology

A contract offers shared domain and information, which helps agents understand each other. Agents can differ in their understanding of their environment, in their goals, and their capabilities, but they can still inter-operate in order to perform a task. The inter-operation among agents is the result of reaching an agreement based on shared understanding, mainly obtained by the reconciliation of the differences. This kind of reconciliation might be accomplished by merging the ontology to which the agent involved refers. The word ontology first appeared in Aristotles philosophical essays, where it is used to describe the nature and organization of being.

There are four main types of ontologies: domain ontologies that provide a vocabulary for describing a particular domain, task ontologies that provide a vocabulary for the terms involved in a problem solving process, meta-ontologies that provide the basic terms to codify domain and task ontologies, and knowledge representation ontologies that capture the representation primitives in knowledge representation languages [43].

Slad and Bokma described the use of ontologies to facilitate the collaboration within enterprise [115]. Jones and Ivezic [56] described the environment required to achieve SN self-integration, this environment should allow for semantic querying, semantic mapping, and semantic inferencing. Smirnov and Chandra [22] described the elements of a general methodology for utilizing ontologies in knowledge management for the co-operative SN configuration. SN configuration entails managing the supply chain knowledge, modeling the constraint network, and managing knowledge among agents. Modeling coordinated SN requires specifying the following concepts: activity, process, SN processes and communication.

Contract: a critical view

Supply network formation using contract is simple since agent decision space is limited to specific items in the contract. It is also stable since agents have no incentive to deviate from the agreed upon contract. It is distributive since SN harmonization models by contracts usually deal with two independent agents. However, contracts are most effective only when complete information is available and all future contingencies can be accounted for.

SN coordination by contracts usually deals with relatively simple structures, consisting mostly of two independent parties. It requires knowledge of relevant cost and demand parameters in order to achieve perfect coordination. Real-world contracts are generally much more complex, consisting of a large number of interdependent issues. A typical contract might have tens or even hundreds of distinct issues. Even with only 50 issues and two alternatives per issue, we encounter a search space of roughly 10^{15} possible contracts, too large to be explored exhaustively. Moreover, the value of one issue selection to an agent will often depend on the selection made for another issue. Such issue interdependency leads to nonlinear utility function with multiple local optima. In practice, uncertainty with respect to the future makes it impossible to design a contract with all contingencies included. In addition, it is very hard to achieve coordination by setting contract parameters at a single point in time. Finally, the central coordinator agent might become a communication and computation bottleneck or a potential failure for the whole system. For more details about coordination using contracting, see [15, 16, 23, 30].

The contract problem is then seen as a constraint distributed optimization problem solving where a set of variables describing a service are shared among a set of agents. However, because there are dependencies among variable values then, the local optimization process needs to simultaneously satisfy both the local, and interaction constraints since the join solution requires the inclusion of other agents' optimization choices. The local problem of an agent is assumed to be defined where decision variables describing a service and their preferred satisfaction constraints are enumerated. The solution of this local problem is then viewed as an optimization problem constrained by a multi-dimensional system. In order to solve this problem, other coordination means are required such as bidding and negotiation.

2.2.2 Market Mechanisms

Auction

McAfee and McMillan [74] define the auction as "*a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from market participants*". The term *auction* appears to be derived from the *auction*(increase), which seems to imply a dynamic process, where individuals continually change their bids. The term *bid* refers to a bid to buy or offer to sell. Auction is a binary relationship, where the outcome is a deal that binds two agents: the auctioneer and the bidder. Both auctioneer and the bidder use competitive strategy to maximize their own utility function/profit. The auctioneer wants to subcontract out tasks at the lowest possible price, while the bidder who handles the task wants to receive the highest possible payment for doing so.

The application of the multi-agent paradigm to bidding can be viewed from two different points of view. First, the mechanisms of a bidding can be defined as a resource allocation problem to a set of agents. Second, the bidding can be viewed as a process of automatic negotiation, where agents interact in an electronic market environment to trade items. The purpose of the bidding is to provide a better balance of resource by taking into account interactions between the agents participating in the auction. The use of auction mechanism for SN was demonstrated in [3, 78, 105].

Auction: a critical view Auction theory provides some advantages for agent based SN formation for practical and theoretical reasons. The Internet-based distributed electronic marketplace has many characteristics of an economy as well as those of a computational system: bids specify a correspondence between prices and quantities of the resources that the agent offers to demand or supply, and agents make their own decisions about how to bid based on the prices and their own utilities. Auctions are a simple and well-defined mechanism, it provides an effective decentralization of decision making. Interaction is distributive since Markets are naturally distributed and the resources are heterogeneous with different ownerships. Through the use of auction, agents may be coordinated based on much simpler set of policies. Agents need not to interact directly with other agents, know each other's preferences, nor even know of each other's existence. An auction has a minimal communication overheads since it is limited to the exchange of bids. Its strict rules govern the behavior of the auctioneer and the selection of a bidder. The interaction variables are limited to the price and quantity.

Auction results in a mutually acceptable solution for both the supplier and the buyer while the market forces alone decide on the price and the auction termination. Auction leads to Pareto-optimal allocations. An allocation is Pareto efficient or Pareto optimal when no further Pareto improvements can be made. Moreover, the second welfare theorem tells us that any Pareto optimum is in principle achievable via the competitive mechanism, thus under classical conditions (preferences and technologies are monotone, smooth, and convex), auction places a lower bound on the quality of solution (Pareto efficiency), and no upper bound (any social optimum must be Pareto optimal). A common criticism of a state of Pareto efficiency is that it does not necessarily result in a socially desirable distribution of resources, as it makes no statement about equality or the overall well-being of a society, notably, allocating all resources to one person and none to anyone else is Pareto efficient [40, 89] .

On the downside, out-sourcing decision is based solely on pricing where other criteria are not traded off with price, therefore three is a big chance of conflict. Since agents are dealing with only one variable, conflict in agents goals and constraints may results failed bid [28, 110]. Agents are only allowed to exchange proposals. In addition, agents utilities or preferences are assumed fixed [8, 133]. Moreover, there is only one bidder that is selected in the outcome. Other problems may arise such as bidder collusion and lying [105].

Negotiation

SN is required to comply with customer orders even if it may be hard to do so. SN have to respond to the orders quickly and efficiently in the limited time available to fulfill the customers requirements. To coordinate different SN entities and solve problems they encounter, negotiation decisions have been identified as crucial for successful global SN. Negotiation techniques are used to overcome conflicts, and to come to an agreement among agents. Conflict is a consequence of failed coordination interaction attempt e.g., bidding [28, 110]. Conflict corresponds to the terms: inconsistency, contradictory or incomplete knowledge. Since agents are autonomous, they have control over their internal state, and will not simply perform an action because another agent wants them to. It is therefore necessary for the collective to come to some agreement about exactly which course of action they will follow. Such an agreement is reached via *negotiation.*

Negotiation is a basic mechanism for interaction that allows the members in a MAS, who have conflicting intrests/goals to coordinate their actions to find a compromise to an issue [132]. The negotiation are intended to improve the global state of affairs or to achieve individual agent's objectives such as: minimizing the time to find a solution, minimizing the total resources usage, maximizing results' quantity, trying to achieve Pareto efficiency. The negotiation process progresses through a dialogue in which offers and counter-offers are exchanged in a common effort to advance towards a mutual agreement.

The major features of negotiation are: (1) the communication language used by the participating agents; (2) the protocol followed by the agents as they negotiate, which defines the proposals that agents can make as a function of prior negotiation history; (3) the decision process that each agent uses to determine its positions, concessions, and criteria for agreement; (4) a negotiation set, which represents the space of possible proposals that agents make; (5) A collection of agent's strategies within the rules of the protocol, one for each agent, which determines what proposals the agents will make; and (6) the information they know about each other.

Negotiation is a process by which agents communicate and compromise to reach a joint decision on matters of mutual interest, each trying to maximize their own individual utilities. The agents first communicate their positions which might conflict, then try to move toward an agreement by making concessions, searching for alternatives and argue about it. Negotiation may fail if the participants are unable to reach an agreement, due to some irreconcilable differences. For more details about negotiation in SN context, see [55, 59, 68].

Negotiation is either competitive or cooperative. Bargaining negotiation is competitive negotiation by means of resolving a conflict over a single mutually exclusive goal (one dimension) (e.g., bidding operates only within price dimension). Competitive negotiation is a win-lose negotiation. The cooperative negotiation on the other hand, allows agents to negotiate over multiple interdependent, but non-mutually preclusive goals (dimensions), cooperative negotiation is a win-win negotiation [64].

Negotiation: a critical view Through the use of autonomous agents, which negotiate on behalf of their owners, a business can obtain flexibility in prices and goods, and go beyond price competition.

Negotiation processes between autonomous agents are costly in terms of communication and computation overheads. If there is no efficient way to negotiate, the negotiating overhead will become a bottleneck, which will reduce the efficiency of agent especially when the environment is dynamic. Negotiation may fail if the participants are unable to reach an agreement, due to some irreconcilable differences. In order to find solution that meets the needs of both sides, negotiation should run on multiple issues. A multi attribute negotiation might overcome the problem of empty negotiation space but on the account of complexity. If we have a set of attributes $A_1, \cdots, A_m$ where each attribute A_i can take a set of values $a_{i,1}, \cdots, a_{i,n}$, then the size of the space of possible deals is $\prod_{i=1}^{n}(A_i)$. The space grows as the number of attributes and the number of possible attributes values increase. When there is a time constraint, it is infeasible to consider every possible allocation solution.

Supply network formation problemis represented by a multi-linked negotiation problem occurring in a multi-task resources-sharing environment. When an agent needs to negotiate multiple, related negotiation issues with other agents about different subjects, the negotiation over one subject has influence on negotiations over other subjects. Dealing with this negotiation independently and concurrently ignoring their interactions will not lead to find a combined feasible solution that satisfies all constraints without re-negotiation over already "settled" issues.

Dealing sequentially with negotiation one at a time and base later negotiation on the result of earlier negotiation will take up valuable time and therefore reduce time available for solving the problem. This leads to reducing potential solution space specially when negotiation deadline taken into consideration. Additionally, given that the result of later negotiations are uncertain, since agent does not have complete knowledge about other agent's state, makes it harder to evaluate a commitment and thus making it harder to find a local solution that will contribute effectively to a good global solution. Negotiations between human participants, however, are richer and more complex than the mere exchange of quantitative offers and counter-offers. Participants request information from each other, collectively seeking common information, try to persuade each other of contested propositions, and advance arguments for their own offers and against those of others. This richness has been recognized by the use of argumentation in multi-agent systems design. Agents may not only present offers in a negotiation, but also the reasons for the offers, any qualifications of, and conditions on them, and reactions to them.

Game theory-based negotiation assumes that agents have complete information about the environment, consequently capable of providing ranking of all possible deals. Agent preference is fixed during negotiation. Therefore they may fail to reach a deal that maximizes their utility, and they fail to achieve a deal. In Game Theory and similarly heuristic based negotiation models, agents exchange *proposal* and are not allowed any additional information. Agent's preferences over proposal are assumed to be *proper* in the sense that they reflect the true benefit the agents receive from satisfying these preferences, which may not be true. The limitation of Game Theory-based negotiation have lead to the emergence of *argumentation based negotiation*.

Argumentation based negotiation overcome the above limitations by allowing agents to exchange additional information, or to argue about their beliefs and other mental attitudes during the negotiation process in order to justify their preferences, or influence another agent preferences. Changing preferences leads to an expanding in the set of individual rational contract. Consequently, improve the quality of the deal reached.

Argumentation-based negotiation The idea of argumentation-based negotiation is that in a situation of conflict of interest agent negotiates by argument in order to make a deal of the best interest of oneself.

Argumentation is a verbal and social activity of reason aimed at increasing (or decreasing) the acceptability of a controversial standpoint for the listener or reader, by putting forward a constellation of propositions intended to justify (or refute) the standpoint before a rational judge [127]. Argumentation can serve both as a framework for implementing autonomous agent reasoning (e.g., about beliefs and actions) and as a mean to structure communication among agents. As a result, argumentation can naturally provide a means for integrating communication with reasoning in a unified framework.

Argumentation based negotiation in a multi-agent context is a process by which agents attempt to convince one another of the truth of some state of affairs. Argumentation requires a great amount of trust between negotiators to share private information. The process involves agents putting forward agreements for and against propositions, together with justifications for the acceptability of these arguments [1, 53, 97, 98]. Argumentative negotiation is cooperative since both parties confer in order to reach a common objective that leads both side to "Win-Win" situation.

In Summary, argumentation lends itself naturally to two main sorts of problems encountered in MAS [95]:

- Forming and revising beliefs and decisions: Argumentation provides means for forming beliefs and decisions on the basis of incomplete, conflicting or uncertain information. This is because argumentation provides a systematic means for resolving conflicts among different arguments and arriving at consistent, well supported standpoints;
- Rational interaction: Argumentation provides means for structuring dialogue between participants that have potentially conflicting viewpoints. In particular, argumentation provides a framework for ensuring that interaction respects

certain principles (e.g., consistency of each participants statements).

2.3 Discussion

The question here is which interaction method to utilize to fit the dynamism of supply network? Interaction protocol is not only driven by uniqueness of supply network organization, since it blends competitive market and cooperative enterprise, but by agent behavior as it changes from competitive in auction to cooperative in negotiation. According to the limitation discussed in this chapter, there is no appropriate particular approach that adapts to the changes in agents' behavior during supply network formation at any particular situation. Therefore, interaction protocol should accommodate more than coordination method according to prevailing circumstances.

Previous research on dynamic selection of coordination mechanisms [5, 34, 71, 100] does not show the actual decision making process where agents decide whether to switch, which coordination mechanism to use, and how to switch to a different coordination mechanism. Therefore we address this issue in our work by presenting such a process. Thus, our work is more concerned with the flexibility involved in adapting to coordinating agents behavior dynamically during run time, which is the line of research we will discuss in the subsequent paragraphs.

The proposed multi-agent coordination approach we aim to propose in this thesis should be flexible enough to adequately address resource constraints (communication overhead, computational and temporal dimension versus value of coordination). Our main argument is that it is preferable not to choose a specific interaction mechanism for each agent at design time, but to develop an adaptive interaction mechanism that is agent selective during run time. Confronted with different situations, an agent can dynamically change the interaction mechanism according to his strategy, and ultimately, increase supply network performance through improved coordination.

Chapter 3

THEORETICAL DEVELOPMENTS

" Mankind are animals that makes bargains, no other animal does this - no dog exchanges bones with another." Adam Smith [116].

In this Chapter, we present theoretical background issues necessary for designing SN interaction protocol and propose one that meets some basic desiderata. In Section 3.1, we introduce agent-based supply network architecture and we formally define SN formation problem. In Section 3.2, we present our proposed three-steps solution to SN formation. Finally In section 3.3, we discuss in detail our adaptive interaction protocol for the three-steps solution.

3.1 Multi-Agent System Architecture

Multi-agent system is an interdisciplinary field, it is inspired by many other fields such as economics, mathematics, philosophy, sociology and ecology. To design a MAS, we need to implement micro and macro designs [135]: During the *micro design*, we think about how to build agents capable of independent and autonomous action so that they can successful carry out the tasks delegated to them. In *macro design*, the focus is on society design and the interaction capabilities (cooperation, coordination and negotiation) in order to successfully carry out the delegated tasks, especially when some conflicts arise in agents goals.

3.1.1 BDI Agent Mental State

The mental state of a BDI agent at any given moment is a triple $< B, D, I >$. Agent uses set of rules to represent beliefs, desires, intentions and Preferences. On the basis of these rules, agent will plan and schedule the actions needed to perform task(s) assigned to it. The action selection function is defined as mapping mental states to actions $\hat{S} \longrightarrow Ac$, where $\hat{S}$ is the set of all agent internal mental states, and Ac is the set of all actions.

Perception

In order for software agent to reason about environment and take an action, it must be able to perceive the state of the world around it. Perception is the information available to the agent about its environment. Any representation of the environment that the agent creates must be derived from its perceptual inputs. Let ρ be the set of all percepts $\{\rho_1, \rho_2, \ldots\}$, a perceptual function maps environment states to percepts, *percept* : $E \longrightarrow \rho$. The mental state of the agent is then updated $\hat{S} \times \rho \longrightarrow \hat{S}$.

Belief

Agent's beliefs are its current knowledge and rules about present environment state, the other agents, and itself. An agent belief gives the agent the ability to accommodate to different circumstances. An agent's beliefs are described as a set of sentences in a formal language together with a deductive process for driving the consequences of those beliefs, it is useful in situations of incomplete knowledge or limited computational resources. It is the basis for all of its argumentation, reasoning, planning and subsequent actions. Because beliefs cannot be perfect, an agent can review its Beliefs and respond with revised plans if necessary, as environmental parameters change. Let B be a set of beliefs, on the basis of the current beliefs and current percept ρ, a *belief revision function* (brf) determine a new set of beliefs, brf:$\wp(B) \times \rho \rightarrow \wp(B)$, and $B\colon = \text{brf}(B, \rho)$.

Desire

Agent's desires reflect the *goals* directed behavior in SN entity. Desires allow agents to dynamically evolve by changing their activity while maintaining coordination across range of situations in the local domain. Desires represent goals to be accomplished, possibly each goal is associated with priority/payoff. An agent's desire represents states of affairs that the agent would, in an ideal environment, wish to be brought about. Desires are assignment of goodness ($\text{value}_{ag_i}(e_u)$) to state of the environment, from agent's perspective. Each agent ag_i has a set of desires $D_i \subseteq D$, Where D is the set of all possible desires. These desires are formulae in propositional logic. A state e_i of environment satisfies a desire d_i if $e_i \models d_i$, where $\models$ is an appropriate semantic entailment relation.

Goal

Agent's goals are the objectives it wants to achieve. A Goal is defined as being top-level task, comprised of a list of tasks that need to be executed by one agent. The goal of agent $ag_i \in Ag$ is a closed formula in first-order logic (i.e., no free variables). An Environment state e_i will be said to *satisfy* the goal g_i if $e_i \models g_i$. A Goal represent rules containing information about future, represented by a finite set of propositional formulas, and achieved via interleaved phases of planning.

Intention

An agent's intention $i \subseteq I$ reflects the reasoning behind taking decision about possible plans for accomplishing a task. Therefore, if one plane fails, another plan is dynamically issued in order to fulfill the desired goal.

Plan

In most BDI systems, the *plan* function is implemented by giving the agent a *plan library*. Plan π is a set of elementary actions $ac_i \in AC$ executed to realize some task within an SN, requiring time and resources for its execution. The realization of plan provokes a change of the state of the environment.

The atomic actions set $Ac = \{ac_1, ac_2, \cdots, ac_n\}$ are the basic steps an agent can perform using some resources. In other means, an *atomic action* is an *elementary partial plan* that can be achieved directly using a resource. Action specifies how the change in the environment state is made. An action moves the environment from state e to another $e\prime$; hence it is a function $ac : e \rightarrow e\prime$. These actions and resources do not necessarily belong to its own set of actions and resources, and therefore an agent may depend on others in order others in order to carry certain plan.

Definition 3.1.1 *A plan π is a set of actions $ac_i \in Ac$ that transforms an input state into an output state under some condition. Input and output states are defined by the state of system $e_i \in E$ used or produced by the plan. Plan can be expressed as transfer function: $e_u = \pi(e_0, e_1, \cdots, e_m)$ subject to $B(c_1, c_2, \cdots, c_n)$. Where $c_1, c_2, \cdots, c_n$ are conditions that can be expressed in the form of a logical predicate, e_u is the output of π, $\{e_0, e_1, \cdots, e_m\}$ are the inputs and B is a Boolean expression linking predicates and represents a set of pre-conditions, which must be verified before the plan π can be executed.*

A plan library is a preassembled collection of plans, in which an agent find a plan to achieve a goal. In implemented BDI agent, pre- and post -conditions are often represented as a list of atom of first-order-logic, and beliefs and intentions as ground of atoms of first-order-logic. A *pre-condition* defines the circumstances under which a plan is applicable. A *post-condition* defines what states of affairs the plan achieves. Finding a plan to achieve an intention is reduced to finding a plan whose precondition matches the agent's beliefs and whose post-condition matches the intention. Different *candidate plans* are generated to achieve a particular set of desires.

Agent try to minimize the amount of resources needed to perform a task, and to schedule its related plans and actions during run time in the most optimal way in order to successfully complete it by its deadline. Plans and actions can be characterized by two parameters, time and cost: Time is used to represent the duration of an action or a plan, it can be represented by a scalar function as follows: Let $ac \in Ac$ be the set of actions, and $\mathbb{R}^+$ the set of positive real numbers. The duration of an action $ac \in Ac$ is denoted by $t_{\text{execution}} : Ac \rightarrow \mathbb{R}^+$. Cost is a measure of amount of resources used for an action or a plan to perform or delegate a task. The function $Cost : Ac \rightarrow \mathbb{R}^+$ returns the cost of using resources for a particular action ac_i. Usually, it is not possible to know exactly the duration and the cost of an action in advance. The duration can be defined by its mean value $\mu(t_{\text{execution}}ac)$ in deterministic case, or it can be bounded by its lower and upper duration values $\{min(t_{\text{execution}}ac), max(t_{\text{execution}}ac)\}$ in stochastic case. Similarly, cost can be defined by its mean values, or by it can be bounded by its lower and upper cost values. However, it should be noted that cost may depend on the duration of the action. The cost function may have parameters other than the plan's action, such as the initial state and domain dependent variables.

Agent's Preference

Decision theory (a mathematical theory of rational decision making) defines a rational agent as one that *maximizes expected utility.* The rationality of agents is represented in terms of some form of utility functions and preference relationship which enable agents to evaluate and compare different alternatives. A utility function is to make rational decision on selecting tasks according to certain constraints and preferences, where preference among alternative is represented by the relation $\underset{i}{\succcurlyeq}$. A utility is a numeric value representing how valuable the state is. The utility of an agent ag_i is often described in terms of a *utility function* u_i:$\Psi \rightarrow \mathbb{R}$, which assigns a real number to each possible alternative that captures the levels of satisfaction of an agent with particular task.

In order to get the agent to do the task, we need to define task specifications in terms of reward and cost in order to get the agent to do the task. A task specification would be associating a task to utility value related to its related environment state, and associating a cost value of reaching task related state. Let $E = \{e_1, e_2, \ldots\}$ be a set of outcomes of plans or states that agent have preferences over. The utility of an environment state $e_i \in E$ for the agent ag_i be given by a function u_i:$E \longrightarrow \mathbb{R}$, where $E = \{e_1, e_2, \ldots\}$ is the set of environment states. The goal of the agent is to strive to bring about the state of environment with the greatest value: i.e., for an agent ag_i $e_0 \underset{i}{\succcurlyeq} e_u$ iff $u_i(e_0) \geq u_i(e_u)$.

A *one-agent-plan* $\pi_i = \{ac_1, ac_2, \cdots, ac_k\}$ is said to move the world from state e_0 to state $e_u \in E$. There exists a *cost* function $\text{Cost}(\pi_i) = \Sigma_{k=1}^{n}\text{Cost}(ac_k)$ over one-agent plans; The goal of the agent is to try to bring about the desired state of environment $e_u \in E$ by choosing the best plan, where $E = \{e_1, e_2, \ldots\}$ is the set of outcomes of plans and the states that agent have preferences over. The best plan π^* is the minimal cost plan: $\underset{e_0, e_u \in E}{min} \text{Cost}(e_0 \xrightarrow{\pi_i} e_u)$ that moves the world form state e_0 to the state e_u for agent ag_i.

In order to analyze how agent might be expected to behave, we need to specify how they are motivated. It is crucial to define the payoff for each agent in order to choose certain plan. Payoff function is an assessment criteria that assign a utility value for each state it desires to reach and cost value for the plans it requires.

Definition 3.1.2 *Payoff of plan* π *for an agent* ag_i *that moves the environment state from* e_0 *to* e_u *is defined as:*

$$Payoff_{ag_i}(\pi, e_0, e_u) = u_i(e_u) - Cost_{ag_i}(\pi_j, e_0, e_u) \tag{1}$$

Where $u_{ag_i}(e_u)$ *denotes worth of environment state to agent* ag_i*,* $Cost_{ag_i}(\pi_j, e_0, e_u)$ *denotes the cost of the plan, and* $e_u \models \pi(e_0)$ *denotes executing plan* π *results moving the environment form* e_0 *to* e_u*.*

Payoff function captures the degree of their willingness to act on a plan. The goal of the agent is to bring about the state of environment with the greatest payoff value, by the choosing the plan between within a set of plans that maximizes payoff value. Thereby the best plan for an agent ag_i that moves the environment form state e_0 to state e_u is a plan $\pi_i^*(e_0, e_u)$ such that $\text{Payoff}_{ag_i}(\pi_i^*, e_0, e_u) \geq \text{Payoff}_{ag_i}(\pi_i', e_0, e_u)$ for all $\pi_i' \neq \pi_i^*$.

3.1.2 Agent-Based Supply Network: Definition

Agent-based supply network (ABSN) is a system of inter-dependent agents from different group sets (markets), where individual agents compete to become partners of ABSN, interact in order to achieve tasks they are committed to. Through their striving to maximize their own goal they maximize ABSN global goal. ABSN is a dynamic organization, where the organizational structure of the system is changing over time, with agents entering and leaving. This adaptive organization motivates our approach to dynamically reallocating tasks and resources in MAS.

ABSN can be defined as a graph G that expresses the organizational structure of SN. This graph comprises a finite set of edges E and nodes N . Edges represents information flow and task inter-dependency relationship. Nodes represents roles that represents the functions of the system, that to be designated by candidate agents. Horizontally SN is made from peers from similar markets. Vertically SN is made of precedent and subsequent agents. SN head has only precedent agents, and SN tail has only subsequent agents. An SN participant in the middle level of SN changes roles between precedent (the customer) to subsequent (the supplier) [60]. Let $\hat{c} \subseteq C$ where C is set of end customers, $\hat{mp} \subseteq MP$ where MP is the set of participant in the middle level, and $\hat{s} \subseteq S$ where S is set end suppliers, then SN is formed by $\hat{c} \cup \hat{mp} \cup \hat{s}$.

ABSN can be seen as a complex dynamic system made of competitive and/or cooperative concurrent agents, where each agent processes some activities that is triggered by some environment events. Agent-based supply network has a goal (or a set of goals) that is/are achieved by a set of *tasks* $T = \{t_1, \cdots, t_n\}$ that are accomplished by a set of *roles*

$R = \{r_1, \cdots, r_n\}$. A set of agents $Ag = \{ag_1, \cdots, ag_n\}$ are required to meet those roles. A role requires a certain set of *skills* that are necessary to execute *plans* π_i, which are a set of *actions* ac_j. The agent that is willing to fill the role and meets the skills requirement become a candidate for an organizational entity (see Figure 2).

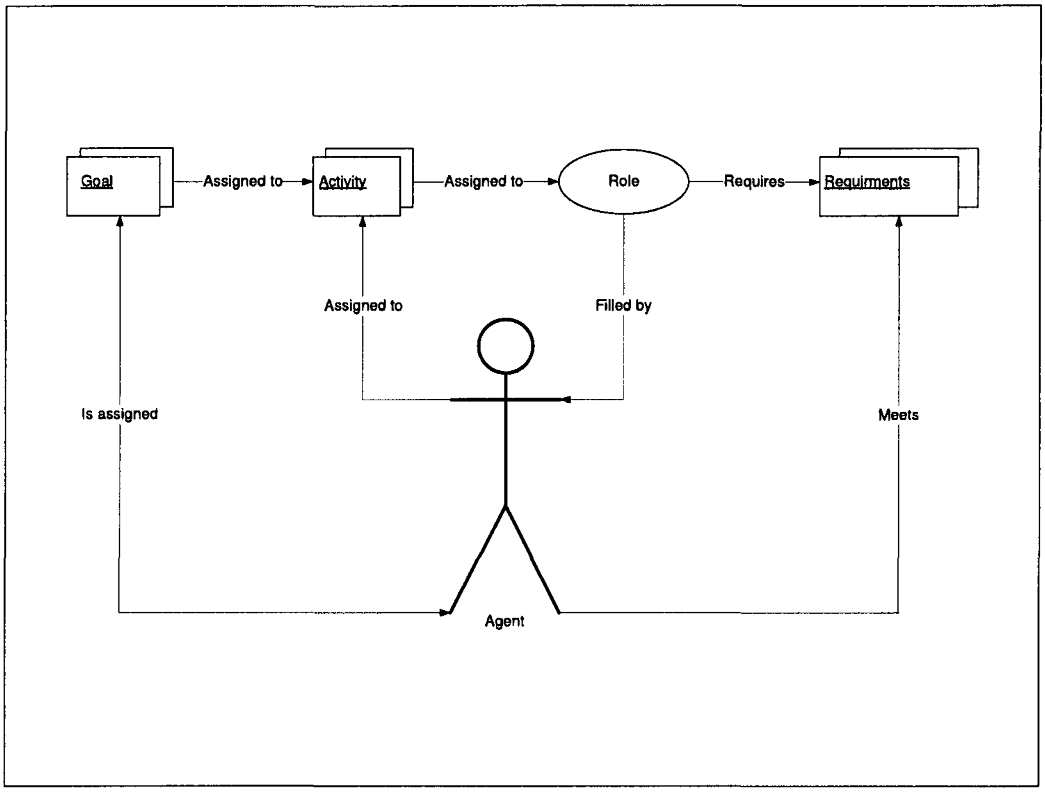


Figure 2: Individual agent role in agent-based supply network

ABSN has a *global goal* that is achieved by a set of tasks that are performed by a set of agents' *roles* that moves the environment from initial state e_0 to the desired state e_u. The global plan comprises a *global schedule* and a set of *joint plans*. A global schedule coordinates the set of joint plans by taking into account their interdependencies. The joint plan $\pi_{ag_i,ag_j} = \pi_{ag_i} \cup \pi_{ag_j}$ is a *joint schedule* and *joint function* between agent ag_i and ag_i.

The joint schedule sc_{ag_i,ag_j} is a partial order over the union of actions $ac_i \cup ac_j$ in the joint plan. The joint function π_{ag_i,ag_j}:$e_0 \rightarrow e_u$ executed by both agents ag_i and ag_j. Where

$e_0, e_u \in E$ is the set of all possible environment. When there exists no realizable joint plan $\pi_{ag_i} \cup \pi_{ag_j} \neq \varnothing$ that moves the environment from its initial state e_0 to a desired state e_u, then negotiation space over joint plans is empty. When negotiation space is empty, we call it a *conflict* situation.

Supply Network Formation

SN organizational structure formation problem is similar to coordinated distributed problem solving. SN agents cooperatively try to solve problems by using their local expertise and information to individually solve subproblems; These subproblems solutions in turn synthesized into a global solution. Distributed agents have control of some variables, attempt to optimize their local payoff/utility function. The achievement of the global objective function is characterized by the aggregation of distributed constraint local utility function.

The SN formation problems have input and output: The input is a set of agents controlling some amount of resources and a set of tasks requiring some amount of resources and worthing some utility or payoff. The output solution is a network that assigns a set of agents to a set of task. SN formation problem refers to the process of configuring the organizational parameters, which are the property of the organizational entity in order to find such an organizational structure $os^* \in Os$ that fulfills SN global goal. This optimal organizational structure that maximizes the social welfare of SN agents set Ag is represented by:

$$os^* = \underset{os_i \in Os}{argmax}\, U(os_i) \tag{2}$$

Where

- $Os = \{os_1, os_2, \cdots, os_n\}$ is a set of possible organizational structures (networks);
- $os_i = \{oe_1, oe_2, \cdots, oe_m\}$ is a task assignment vector of a set of organizational

entities (nodes);

- The global utility $U(os_i)$ is the sum of the utilities on all the nodes $U(os_i) = \sum_{oe_j \in os_i} u(oe_j)$;
- $oe_j =< ag_j, r_j, t_j >$ is the organizational entity that is to be filled by an agent;
- $u(oe_j)$ is the utility function for each organizational entity.

Task Delegation Supply network is a network of agents (nodes) working together to solve problems that are beyond their individual capabilities. As agents have different capabilities, if an agent does not have the resources to execute a task on its own and the task is decomposable, agent breaks down the task into sub-tasks, and delegate it to a group of agents. If the agent does not have all the resources needed to accomplish the task, and the task is not decomposable, then it is said that the agent cannot execute that task. We can define the potential for outsourcing with respect to agent's goal G_i. There exist a potential for outsourcing for agent ag_i if:

1. There is some agents that belongs to different groups (markets), such that agent ag_i believes that these agents can jointly achieve G_i and,
2. Agent ag_i cannot achieve G_i in isolation and,
3. Agent ag_i believes that every action ac_i needed to be outsourced is necessary to achieve G_i.

Partner Selection Since each agent (ag_i) is keen to find its appropriate joint plan partner at each neighboring market, the network formation goal boils down to finding optimal task assignment subset between two partners, such that each task's requirements is satisfied and both agent's utility maximized. Let $u_e(ag_i, ag_j)$ be a utility generated on the edge e between ag_i and ag_j. This represents the reward generated on the edge between oe_i and oe_j, contingent simultaneously on the values of both variables $(u_{ag_i,ag_j}, u_{ag_j,ag_i})$ and hence referred to as a constraint. The organizational structure design problem is then simplified into finding the agent that maximizes bilateral welfare. i.e., $\forall\ ag_i$ the problem is to find the candidate agent such that:

$$ag^* = \underset{ag_j \in Ag}{argmax}\ u_e\{ag_i, ag_j\} \tag{3}$$

Then the global SN utility $U(os_i)$ is the sum of the rewards on all the edges according to the assignment of tasks of $oe_j \in os$.

As agents work together, they face two constraints: First, their individual solution to subproblems must be integrated into an overall solution. Therefore, they must coordinate their problem solving efforts using interaction to resolve inconsistent views and reach some agreements. Second, limited communication means that agents must rely on their own reasoning to modify their behaviours as circumstances change. Agents make decisions based on local views that might be incomplete, inconsistent, or out of date. Therefore they must use their communication and computation resources not only to perform tasks, but also to interact to adjust their behaviours.

3.2 Three-Steps Solution: Contracting, Bidding and Negotiation

Network formations problem boils down to task assignment problem, which is further simplified through decomposing it into a set of bilateral joint plan and reaching a bilateral deal between competing/coordinating agents. Hence the bottom line to our problem is to find the best agent. By modeling the real world, the protocol we specify in this chapter is intuitive and offers the basic desiderata that one would expects from an interaction protocol.

Contracting provides task decomposition, an initial proposed solution through centralized planning for distributed plans approach, and commitment in case of joint agreement.

Bidding on the other hand realizes the proposed central plan through searching an agreement space and checking the plan feasibility by applying the proposed plan constraints to SN candidate, and applying candidate's local constraints to the proposed plan. Furthermore, bidding provides further optimization of the proposed plan and narrows down conflicts to certain issues. Finally negotiation is the last resort for agents to reach an agreement. It is applicable when agents fail to reach an agreement due to lack of knowledge about the existing constraints/alternatives. The purpose of multi issue negotiation is to increase efficiency in reaching agreements through expanding the negotiation space set. It is applicable when agents fail to reach an agreement due to lack of knowledge about the existing constraints. In fact, negotiation is used to release constraints when no realizable solution is obtained under auction. Negotiation also serves as a source of knowledge to agents in order to learn, change beliefs and adapt to changing circumstances.

3.2.1 Contracting

Contract initially represent a central planning approach from the point view of the resource seeker in order find a preliminary plan proposal. Contract terms entice individual agents to act in globally desired way by affecting their choice of local optimal control parameters. An agent may achieve its desires by contracting certain tasks to other agents. Agents divide problem (contract) into sub-problems (sub-contract) and assign each sub-contract to an agent role (organizational entity), which is to be filled by a capable agent.

Contract

Since agents are self-interested, they would only perform actions for one another if they receive something in return. For example a *Seeker* is committed to reward a *provider* if it executes the *task*, resulting in achieving their own desires if they get the tasks done.

A *bilateral contract* can be seen as agreed upon specifications of exchange of tasks/rewards between two agents. A bilateral contract $\Omega_{i,j}$ is an expression of the form:

$\text{do}(ag_i, T_i) \;\wedge\; \text{do}(ag_j, T_j)$, where ag_i denotes the agent that executes the task T_i and the same applies for agent ag_j for task T_j. The task $T_i \in \Omega_{i,j}$ denotes agent ag_i part of the contract $\Omega_{i,j}$, and *do*(.) refers to executing a task. Fulfilling the bilateral contract $\Omega_{i,j}$ moves the environment to state e_u, in other mean $e_u \models \Omega_{i,j}(e_0)$.

An economically rational agent accepts only contracts that it perceive as *individually rational contract* (IRC). A contract is an IRC to an agent if that agent is better off with the contract than without it. A bilateral contract $\Omega_{i,j}$ is individually rational for agent ag_i in environment state e_0, if it believes that $\forall\ \Omega_{i,j} \in \Gamma_{i,j}\ u_{ag_i}(\Omega^{IRC}_{i,j}, e_0) \geq u_{ag_i}(\Omega_{i,j}, e_0)$, where $\Gamma_{i,j}$ is the set of all possible contracts between ag_i and ag_j.

If agents do not change their beliefs, then the intersection set $\Gamma^{IRC}_i \cap \Gamma^{IRC}_j$ for agents ag_i and ag_j is the set of possible rational deals. If $\Gamma^{IRC}_i \cap \Gamma^{IRC}_j = \emptyset$, then agents are in conflict, and they will never reach a deal unless they change their beliefs over preferences over deal issues.

In the solution proposed by this thesis, contractual agent use contracts for enabling social control as a joint commitment between agents to form a social relationship and adapt their current and future behavior in accordance with the contract clauses. Contract supports social aspects in automated interaction e.g., roles, and interaction issues e.g., termination time, winner determination, penalties, and decommitments.

A *Contract clause* defines general commitments, which apply to all of the contracting parties. Commitments are viewed as pledge to undertake a specified course of action. Therefore, agent's commitments should be consistent with its beliefs.

Failure by subcontractor to fulfill the requirement of contract e.g., meet the delivery date of task T_i results in penalty $t_{\text{tard}}(T_i) \times$ *tardness*, where $t_{\text{tard}}(T_i)$ is the time by which delivery of product or service is late, and tard is the marginal penalty for missing the delivery date. In order to allow agents to accommodate new events, contract may incorporate decommitment (freed from the obligation of the contract). Decommitment penalty is the penalty the contractor agent has to pay to the other contractor agent if it decommits the contract.

A *deal* δ is an agreed upon joint plan π_{ij} between agents ag_i and ag_j that moves the environment from state e to state $e' \in G_i \bigcap G_j$ that satisfies both agents goals. When both agents are committed to a deal δ it becomes a bilateral contract $\Omega_{i,j}$. A Bilateral contract $\Omega_{i,j}$ between two agents ag_i, ag_j is also an agreed upon and (committed to) joint plan $\pi_{i,j}$. The cost of contract $\Omega_{i,j}$ for an agent ag_i is the cost of its part in the joint plan related to that contract $\text{Cost}_i(\Omega_{i,j}) = \text{Cost}_i(\pi_{i,j})$. A deal is a vector of set of issues $\{A_1, \cdots, A_m\}$, where each issue A_i can take a set of of values $[a_{i,1}, \cdots, a_{i,k}]$. Within the proposed joint plan a rational agent attempts to reach a deal that maximizes its payoff. The payoff for an agent from a deal is simply the difference between his expected reward from the deal and the cost of achieving his part of the related joint plan. A deal is *individual rational* (IR) if all agent are at least as well off with the deal as they were with the initial allocation. A deal δ is *Pareto optimal* if there exists no deal that dominates it (there is no deal that is better for one of the agents and not worse for the others). The set of all deals $\delta_1, \cdots, \delta_n \in \Delta$ that are individually rational (IR) is the negotiation set space. Each agent search space of potential agreements in attempt to reach an allocation which gives it as much as utility as possible, in other mean reach Pareto optimal. If however, bidding fails to find such a possible deal, then a multi issue negotiation will start in order to release the constraints and increases the negotiation space set.

Contract proposal

The plan is decomposed into task. The tasks are announced among SN participant in form of a series of contract proposals that contains goals, constraints, partial plans, roles, ontology, norms, and social law. A *contract proposal* is announced in order to find *interested partners*, the entities that are part of *market groups*, which are interested in joining the SN according to contract terms, role requirement and has the skills, the resources required (organizational entity attributes); And it is willing to bid for the task. A contract proposal is a contract that has been suggested by some participant(s) but has not yet been accepted by parties whom *role* are mentioned in the proposal. After all relevant parties accept contact proposal, it becomes a *contract*. The agents that proposed and accepted the contract proposal are both committed to keep their part of the contract.

3.2.2 Bidding

The purpose of bidding is to ensure that task(s) assignment, which the seeker agent cannot satisfy are re-allocated to the most competitive provider through auction mechanism. Bidding guarantees that the cost of the allocated task from the global plan and assigned to agents is minimized. The system proposed in this thesis acts as a distributed market places where providers are available in each market and auction conducted at each market. Our motivation for avoiding combinatorial auction is the fact that the winner determination problem is an NP-complete, means that it is impossible for us to find a polynomial time algorithm as the solution [106], because of the expected wide variability in problem size, and because of the need to be able to limit the time spent for evaluating the bids.

In this thesis, *Reverse-first-price-open-cry auction* is used. This auction resembles *Dutch auction* in which the auction starts with high asking price, which is then progressively lowered until a bidder accepts the current auctioneer price. Auctionprotocol is similar to Dutch auction, where the sellers (bidders) continually lower their price until the buyer (auctioneer) accepts the product or service at the current price. The difference between dutch auction and reversed first price open cry auction is that in dutch auction there is only one seller agent and multiple buyer agents, however in our case here, there is multiple selling agents and one buyer agent. In addition, dutch auction the seller is the driving force of the interaction, but in reverse auction buyer is the driving force of interaction.

Bidding process

Auction protocol resembles single-issue bilateral negotiation protocol, which is basically an alternating offer from one side. Let ag_s denotes the seeker agent, ag_p the agent who provides the goods/services and let $P_{p,min}$ denotes the minimum price the provider agent is willing to take as a reward in exchange of service. Let $P_{s,max}$ denote the maximum price the seeker agent is willing to pay in exchange of goods/services. A price of a deal that is acceptable to both ag_s and ag_p belongs to the *price agreement interval* $[P_{p,min}, P_{s,max}]$ where $P_{s,max} \geq P_{p,min}$.

The request-for-quote announced should be accompanied by *contract proposal*, which contains task(s) related issues and its value such as time and quality constraints. This information comes from preliminary global plan. Auctioneer (seeker agent) receives and evaluates bids when bids return, and then determine the optimal allocation plan based on the bid prices of the feasible plans related to fulfillment of particular task. The winner is the bidder that offers feasible-plan with the minimum-cost.

After synthesizing the final solution (global plan) successfully, the auctioneer sends bidding price acceptance message to winning contractor, and accordingly a mutual commitment is established and a contract is made, and the *interested providers* who won the bid are selected to a *supply network partners* or an organizational entities. A series of reverse back-to-back open-cry concurrent auctions is instantiated where supply network interested providers bid competitively on price in order to find maximum utilization of resources. If seeker founds a bidder with lowest possible price it means this the lowest possible feasible solution to the task to be delegated, consequently agents mutually agree to commit on contracts.

However, if auction mechanism fails to solicit bidders to cover all the tasks in the call-for-quote, the failure handling process would be invoked. The bid can fail because offers is not acceptable by the perspective agents due to any conflicts that arise because either price is higher than what seeker expect, or because agents object on some constraints in the proposed contract. *Interested providers* are the ones that are considered to negotiate the contract issues since they are *potential supply network partners.* Agents cooperate through multi-issue negotiation is order to relax global plan issues (constraints). Relaxing conflicting constrains expand the negotiation space and increase the possibility of finding a feasible solution. Potential supply network partners suggest counter-proposal, the seeker agent determine the winning contract-proposal through enumerating it payoff function for all the candidate feasible contract-proposal that can cover the task to be delegated. After reaching winning contract-proposal, a new call-for-quote is announced to entice new interested provider from the related market. Re-bidding is invoked in order to find improve the new solution for the unbidden portion of the tasks at certain node, while having one guaranteed feasible solution (the winning proposal).

Bidding strategy

In the proposed solution, a bidding strategy for an agent is a function that takes the last offers made so far, and returns an offer to make next run. Seeker agent strategy dictates when exactly the customer should accept price shouted by seller agent. It is simple and requires acceptance of shouted price that falls below given threshold $P_{s,max}$. Let bp_t be the current bidding price, $u_s(\delta)$ be a seeker utility function, $P_{s,max}$ be the maximum price the seeker agent is willing to pay in exchange of goods/services, t_B be the bidding ending time, and t_A be the auction deadline time. A seeker agents bid strategy $B_s(bp_t, t_B, u_i(\delta), P_{s,max}, t_A)$ defines the bid that an agent is willing to accept in every run. Let t denotes current clock time, the best-response strategy B_s^* is as follows:

$$B_s^* = \begin{cases} \text{Accept } bp_t, & \text{if } bp_t \leq P_{s,max}, \text{and } u(\delta) > 0 \\ & \text{and } \{t \leq t_B \text{ or } t \leq t_A\} \\ \text{refuse}, & \text{otherwise} \end{cases} \quad (4)$$

Provider agent bidding strategy dictates when the the auctioneer agent should provide a quote. Let bp_i be the provider intended bidding price, $P_{p,min}$ be the minimum price the provider agent is willing to take as a reward in exchange of service, $u_p(\delta)$ be the provider's utility of the proposed contract and A_i be the issues of the proposal. A strategy $B_p(bp_i, bp_t, u_p(\delta), P_{p,max}, A_i, t_B, t_A)$ defines the bid that an agent is willing to place for every run. The best-response strategy B_p^* is as follows:

$$B_p^* = \begin{cases} \text{Bid } bp_i, & \text{If } bp_i < P_{p,max}, \text{and } u(\delta) > 0 \\ & \text{and } bp_i \leq bp_t \text{ and } A_i \text{ is feasible} \\ & \text{and } \{t \leq t_B \text{ or } t \leq t_A\} \\ \text{Counter-propose}, & \text{If issues } A_i \text{ is not feasible and } t < t_B \\ \text{Quit}, & \text{otherwise} \end{cases} \tag{5}$$

The search for the improved solution along the curve of feasible solutions is performed by using bidding decrement. Bidding decrement step size is hard to decide; The smaller the bidding decrement step size, the longer time the auction will be, with the higher probability of winning the bid. Consequently the more improved the solution. The larger the bid step size the faster the auction will be, on the account of not winning the bid. Step size can be determined through run time using agent learning algorithm for more details see [10, 12]. In our model the bidding decrement is decided during run time by the agent according to the following equation:

$$\text{Bidding decrement} = \frac{\text{price gap}}{\text{time left for bid to end}} \tag{6}$$

Where price gap is $\max\{0, \text{current bidding price} - \text{reserved min. price}\}$. Since agents are using homogeneous strategy, the agent with the lowest cost will be eventually the winner.

3.2.3 Negotiation

Negotiationis a process that takes place between two or more agents who are attempting to achieve goals that are conflicting. If participant collectively have objections to some constraints, it is necessary for the collective to come to some agreement about constraints, such an agreement is reached via negotiation. Negotiation is used instead of solving such a computationally NP_hard problem. When bidding fails, information is revealed in order to find new negotiations space. Agent may persuade each other about the desirability of a particular proposal by introducing new issues, or changing some issues in the proposed solution. Agents continuously exchange their information about goals by communication until they reach an agreement about the issues they are in conflict, in order to modify the original plan to find a feasible solution for the plan by releasing constraints or further optimization, after updating primary plan and finding a solution, a new contract proposal is announced and another is bid is executed to find best matching candidate, the winning participant declare their commitment by singing on bilateral contracts. In the framework proposed in this thesis, negotiation can be seen as the a process of joint search through the space of all deals (proposed contracts) $\Gamma_{i,j}$, in attempt to find acceptable contract $\Omega_{i,j}^{IRC}$ (mutually IRC). In fact, negotiation is achieved through the exchange of illocutions in a shared communication language. The exchange of illocutions is driven by the participants agents' needs and goals. This exchange is subject to interaction protocol and *shared conventions* on the intended usage of the illocutions.

Proposal

A *proposal* is some kind of solution to the problem that the agent face. It is an intention to perform a joint plan. It may be a single complete solution, single partial solution, or a group of complete or partial solution. Proposals can be more complex than suggestions or joint action. They may include trade-offs or suggest conditions under which the proposal holds.

Counter-proposal

Agent may respond with *Counter-proposal.* A counter-proposal is just a proposal, which is made in response to previous proposal. The counter-proposal amends part of the initial proposal (which is more favorable to the responding agent than the original).

Conflict resolution

A *conflict* situation is one in which the negotiation set is empty (no individual rational deals exist). The conflict occurs when the agents cannot reach an agreement. A *non-conflict* situation is the one in which there exists a deal in the negotiation set that is preferred by an agent over achieving his goal alone. For negotiation set to be non-empty there should be no contradiction between two agents' goals i.e., $G_i \cup G_j \neq \varnothing$. In case of bidding failure, agent reason about issue to change each other preferences over goals' issues.

Multi-issue negotiation process

It is possible to see the dialectical process as a search process oriented towards finding a situation suitable for both parties. In the proposed framework, in order to find solution that meets the needs of both sides, negotiation should run on multiple issues. The multi-issue utility function approach allows agent to associate a value $v_{i,k}$ and a weight $w_i \in \mathbb{R}$ for each issue A_i among the n issues. This approach presents a quantifiable measurement to evaluate each deal δ for agent ag_i in a single equation:

$$u_{ag_i}(\delta) = \sum_{i=1}^{n} w_i \sum_{j=1}^{m} v_{i,j}(A_i) \tag{7}$$

A multi attribute negotiation might overcomes the problem of empty negotiation space but on the account of complexity. If we have a set of attribute $\{A_1, \ldots, A_n\}$ where each attribute A_i can take a set of values $\{a_{i,1}, \ldots, a_{i,m}\}$, the negotiation space set Δ will contain $\prod_{i=1}^{n}(A_i)$ possible deals. The space grows as the number of attributes and the number of possible attributes values increase. When there is a time constraint, it is infeasible to consider every possible solution. Therefore we use negotiation only in order to reach the feasible solution area through releasing conflicting constraints of issues. The counter-proposal are valued through enumeration of multi-issue utility function and the best feasible solution (best deal proposal) δ^* is the solution that gives the highest utility or payoff value after releasing the constraints.

$$\delta^* = \underset{\delta_i \in \Delta}{argmax}\ u_{ag_i}\{\delta_i\} \tag{8}$$

Negotiation strategy

Let $u(\delta) = f(\alpha u(A_x), \beta u(A_y), \gamma u(A_z))$ be a utility function of a deal, Cost(δ) = $Cost(A_x, A_y, A_z)$ be cost of a deal, where proposal issues are A_x, A_y, A_z respectively with importance factors α, β, γ then the payoff function of a deal δ is:

$$Payoff(\delta) = Utility(\delta) - Cost(\delta) \tag{9}$$

For every agent there is a payoff function with reserved value. The provider agents suggest counter-offer with altered issues values and reasons supporting its position in order to change the importance of weighting factors or release constraints. The negotiation continue by exchanging counter-proposals, and each counter-proposal is evaluated through payoff function enumeration. The winning proposal is the proposal that provides the seeker agent with the highest payoff value. The winning counter-proposal is considered as one feasible solution. The buyer will seeks further improvement to the winning feasible solution by announcing a new bid.

Let $u_i(\delta)$ be the agent ag_i utility or payoff of his contract proposal, $u_i(\delta')$ be utility or payoff of the counter-contract proposal, $u_i(\delta^{IR}_{min})$ be the minimum reserved payoff of any individual rational deal, Δ' be other the counter-proposals suggested by other agents, A_i be the deal related issues and t_A be the auction deadline time. A strategy

$N(u_i(\delta), u_i(\delta'), u_i(\delta^{IR}_{min}), A_i, t_A)$ defines the proposal that an agent ag_i is willing to place and the counter-proposal is willing to accept. The best-response strategy N_i^* is as follows:

$$N_i^* = \begin{cases} \text{Propose } \delta, & \{\text{If } A_i \text{ is not feasible, or } u_i(\delta') < u_i(\delta^{IRC}_{min})\} \\ & \text{and } u_i(\delta) \geq u_i(\delta^{IRC}_{min}) \\ & \text{and } u_i(\delta) \geq u_i(\delta') \\ & \text{and } t \leq t_A \\ \text{Accept } \delta', & \text{If issues } A_i \text{ is feasible} \\ & \text{and } u_i(\delta') \geq u_i(\delta^{IRC}_{min}) \\ & \text{and } u_i(\delta') \geq u_i(\Delta') \\ & \text{and } t \leq t_A \\ \text{Quit,} & \text{otherwise} \end{cases} \tag{10}$$

Consider the following dialog, in which we assume that the buyer proposes a contract that he pointed down to a quality level of 6σ and a lead time of 4 weeks and asking for a price of a deal of 100 pieces, and the auction mechanism fails to find an offer:

- ag_{Seller}: I offer you the goods if you reduce the quality to 4σ and increase the lead time to 6 weeks
- ag_{Buyer}: I reject! I abide by deadline and I need high quality goods, it is the most important issue!
- ag_{Seller}: Why do you need a level of 6σ?
- ag_{Buyer}: Because I do not want to lose money in damaged items
- ag_{Seller}: I cannot offer 6σ, but if you accept a 4σ quality level, I'll change the damaged items; quality is less important than lead time!
- ag_{Buyer}: I agree, come and bid again!

3.2.4 Global Plan Solution Synthesis

Task and allocation mechanism needs to be established from the bottom up, meaning that every participating agent makes individual decisions based on local knowledge and preferences while considering the global constraints, without regard to global efficiency. The global efficiency is generated from the bottom up through bilateral interaction among self-interested agents. The seeker agent receives a solution to all of the subtasks it delegates, and synthesizes an overall solution. Upon formation of SN, the agreed upon plan is executed by the joint action of agents. During this stage, agents interact to execute a joint plan, that they believe will achieve the desired goal. Configured SN structure is represented by a constellation matrix [42], where the column's element represents different market groups. Each cell take has a vector values $\{A, P, S\}$ where $A \in [0, 1]$ indicates if the entity is an active partner. $P \in Ag_1, \cdots, Ag_n$ indicates precedent partners and $S \in Ag_1, \cdots, Ag_n$ indicates subsequent partner. During operational re-configuration, the constellation matrix size is always changing based on varying capacity utilization requirements.

Agents track the execution of the plan through examining the performance against the contract *norms*. When a discrepancy between the plan and actual situation is perceived, a plan control is maintained through re-negotiation of existing commitments, re-bidding portions of the plan, re-planning for jeopardized goals, or abandoning sub-goals that is not needed any more [42].

3.3 Adaptive Interaction Protocol

In our context of SN formation, coordination involves the inter-working of a number of agents, subject to a set of rules. The specification of what is possible in particular coordination context is given by the interaction protocol. Thus, such protocol indicates the roles that are involved in the coordination activities, what communication flows can occur between these parties, and how the participants can legally respond to such communication.

Agent interaction protocol defines the behavior of a group of interaction agents. It describes communication pattern through defining: (1) allowed sequence of messages between agents having different roles; (2) content of the messages; (3) semantics that are consistent with communication acts. Instead of making agent interaction behavior dependent on interaction protocol, we propose the idea of adapting the interaction protocol to the agent behavior. The design of protocols that elicit the agent's preferences is a key problem to social interaction. To gain flexibility, we combine the negotiation protocol with the market rule based protocol. Through iterative multi stage bidding and negotiation, agents assess how their choices of actions affect subsequent agents and get affected by precedent agents. Through the iterative multi stage interaction, agents exchange enough information to find configuration that satisfies the global constraints in addition to their local constraints, instead of having a global view of all agents choices. Since MAS-based SN has a limited number of interaction, it is appropriate to define interaction per-formatives by describing agents possible responses according to our interaction protocol. In this way, it is easier to verify the sufficiency of the per-formatives through modeling the protocol using certain formal tools such as interaction diagram and finite state machine (FSM).

3.3.1 Interaction Diagram

Interaction diagrams (sequence diagrams, and collaboration diagrams) are used to define the behavior of groups of agents. Agent-based unified modeling language (AUML) [88] is used to model conversation, sequence diagram in (Figure 3) captures the chronological sequence of communications between two agents. The right dotted box in the upper right indicates that this a template with unbound parameters. It is divided by two horizontal lines into three categories: (i) role parameters, (ii) constraints, and (iii) communication acts.

The communication acts are represented by the arrows labeled with message names. The agent lifeline in the interaction diagram defines the time period during which an agent exists, represented by dotted vertical lines. The lifeline starts when the agent of a given role is created and ends when the SN life cycle ends. For each agent, there are a number of vertical activation bars. Each bar represents a different agent role or a different processing thread of the agent. The separation of these bars allows an agent to have different lifelines. Since the behavior of the agent role depends on the incoming message, the life line split up in order to describe the different reactions of the agent depending on the incoming message. Life line is split into two or more lifelines to show 'AND', 'OR', and 'XOR' decisions, corresponding to branches in the message flow. Decision graphical connector types are either horizontal line with no diamond symbol depicts 'AND', means all communication acts are sent concurrently. An empty diamond symbol depicts 'OR', means one or more messages are sent. Finally diamond symbol with 'x' in it indicate a decision 'XOR' resulting zero or more messages being sent. the lifeline of an agent role is split accordingly into two or more lifelines and the *threads of interaction* along the lifelines define the reaction to different kinds of receive messages.

Proposed adaptive interaction protocol

The SN organizational structuring process can be considered as a set of bilateral interactions between agents.This process can be considered within the context of electronic market as follows (Figure 3):

- When an agent that receives request for a task, first it determines whether or not it has the capability to carry out the task of such type within the deadline.
- If the task is not atomic, and there is subtask it does not have the skill to do, it needs to delegate this task to its subsequent agents (providers).
- Seeker agent generates a proposal contract for others to carry out the task, and sends an announcement message to all in its subsequent agents' market denoted by(CFQ $\{ag_1, \cdots, ag_n\}, O, I, R, t_B t_A$) where O denotes portable

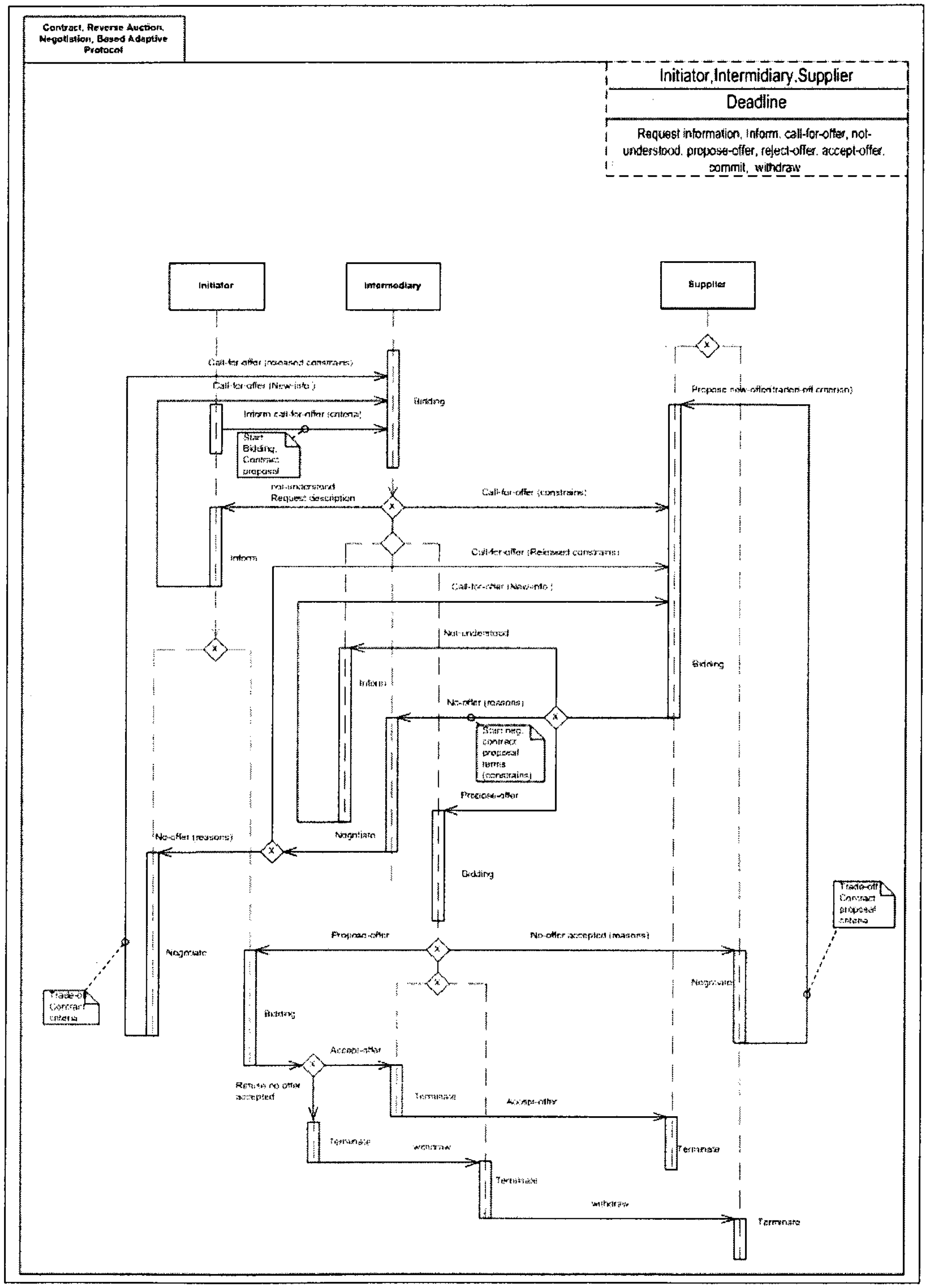


Figure 3: Interaction diagram

shared ontologies needed for interaction, R denotes required roles and I denotes issues variable, t_B denotes bidding deadline and t_A denotes auction deadline time.

- Each time a new bid bp_t is offered it is announced to auction participants (*CFQ(new-price)*), the seeker agent (auctioneer) waits until a given bidding deadline t_B to see if any participant signal their willingness to handle the task at their proposed bidding price bp_i. If a participant does not understand the ontology or syntax of the *CFQ*, it replies (*not-understood*) communicative act. As soon as participant offers a new bidding quote bp_i, the auctioneer issues a new call for quotes (*CFQ(new-price)*) with the decremented price bp_t. The bidding continues until no auction participants are willing to bid, or the auction deadline is reached. At this point the auction ends. If the last quote offered by a participant is below auctioneer's reserved price, the task is delegated to that participant for the agreed quote, otherwise the auction fails (*withdraw*). The participants are informed about the end of the auction and both seeker and provider are committed to the deal (*accept offer*), and the proposal becomes final contract.
- Interested agents respond by filling their values for the required attributes. The matching is based on the attributes of the role. The requirements of the role(s) are structured into skills availability and capability (organizational entity).
- If the interested agent meets the requirements for the role, this agent becomes a potential team partner.
- If an agent is interested in performing any of the roles, it requests for more information about a specific role (*request description*).
- Participating agents propose a bid (*propose offer(bp_t)*), where the content of the bid corresponds to the requirement expressed in SN bid announcement (proposal contract and Bid instructions).

- Customer agent announces a call-for-quote CFQ attached with contract proposal to the preceding providers market agents that are interested in becoming partners of the SN. The customer agent informs all participating agents that the auction has started represented by the message (*inform(start-auction, proposal contract attributes)*) and announces the details of the proposed contract.
- Subsequent agent (e.g., manufacturers, distributers) do a similar process to what initiator did. Subsequent agent analyzes the required tasks, and generates a preliminary solutions (another contract proposals) that reflect its local constraint in addition to the original constraints. If the subsequent agent finds that there is no enough resources available to meet the requirements of the tasks, then it in turn, asks for the help of its subsequent agents, by delegating these unsatisfied task(s) to others, and announces CFQ.
- Agent announces the preliminary solution within contract proposal clauses in terms of quality's upper boundaries (lower specification limits) or σ level, delivery time deadline, penalties on tardiness and decommitments, product structure, bill of materials, specific ontology, and role requirement.
- After the interested partners receive the requirements of the roles, they acts as providers, decompose task if necessary, and start to perform reverse back to back auction for the task they are unable to fulfill. After they receive quotes from their subsequent agents, they evaluate them in terms of cost, quality, and time constraints, and return their quotes if feasible to their precedent agent(s).
- Task decomposition would include further decomposition and sub-subtask assignment, recursively to the point where task becomes atomic and the agent can accomplish it alone. For each stage, a task is decomposed further into a new sub-task. A new "proposed sub-contract" is issued that reflect the original constraint, in addition to the current local node constraints. This way guarantees constraints propagation throughout all SN echelons' markets.

- Once an agent receives the bids from its subsequent agents, it will include it in its own utility/payoff function model, check its feasibility and announces its bidding price for its precedent agent(s).
- The providers continuously lower their prices until seeker takes the item at the lowest possible current price.
- Once the seeker (auctioneer) agent receives the bid, evaluate them based on payoff function and select lowest bid that satisfies contract proposal clauses, then the auctioneer rewards the contract to the winning bidder.
- After the auction end successfully, winning agents are notified of their commitments, and become members (node) of the supply network.
- If auction is successful on agreed upon terms, supply network formation is successful and ready for respond to the initiator's demanded task(s).
- If two entities have the same bid winning price, an announcement that two bidder won the bid is sent out and the winner is determined either by flipping a coin, or by further negotiation to find better terms.
- If during the auction process, no bid exists (*No offer(reasons)*), or non of the bidding prices is accepted (*No offer accepted (reasons)*). The agents may negotiate with each other if required.
- During negotiation providers agents argue to release constraints by making a counter-proposals (*propose offer*). Bidding participant agents propose a counter-proposal with amended issues supported by some reasons (arguments). Auctioneer agent reason about their argumentation (e.g., higher quality vs. less quality and more damaged items vs. compensation for damage items) and calculate the utility and the cost function and prioritize counter-proposal according to their payoff for those counter-proposal. The seeker agent evaluate those counter-proposal on the basis of payoff function and on the basis of quality, price, and time constraints, if possible it releases constraints; Other wise it

reject it *(reject offer(reasons))*

- After reaching an agreement about negotiation issues (e.g.,price, quality, and time), proposed contract terms are altered accordingly. agent chooses the winning proposal to be its primary feasible solution.
- Once an agreed upon proposal is reached, a new cycle of bidding is initiated in order to improve the solution, taking into account the existence of at least one solution, which belongs to agent who performed the negotiation. Agent announce the new proposed contract in order to find more improved solution, and repeat the bidding process until a mutually agreed deal is reached.

3.3.2 State Transition Diagram

Agents can perform various roles and can be in different states within each role in the interaction protocol e.g., manufacturer agent act as a seller in an auction between manufacturer and customer. At the same time, a manufacturer can act as a buyer in another back-to-back auction i.e., in an auction between manufacturer and supplier.

We use state transition diagrams to model the behavior of our agents within the SN system. the idea is to define all possible states an agent can reach, and how an agent state changes. they are suited for defining the behavior of one single agent, however they are not appropriate to describe the behavior of a group of interacting agents.

We can specify our protocol as a form of state transition diagram which gives the various legitimate states that an agent may be in during an interaction and thus the legal transition between states which an agent is allowed to take. The finite state machine (FSM) diagram in (Figure 4) illustrates the internal processing of agents performing their roles.

Finite state machine is a model of behavior composed of a finite number of states, state transitions between those states, rules or conditions which must be met to allow a state transition, and input events which are either externally or internally generated, which may possibly trigger rules and lead to state transitions. The process begin when seeker agent

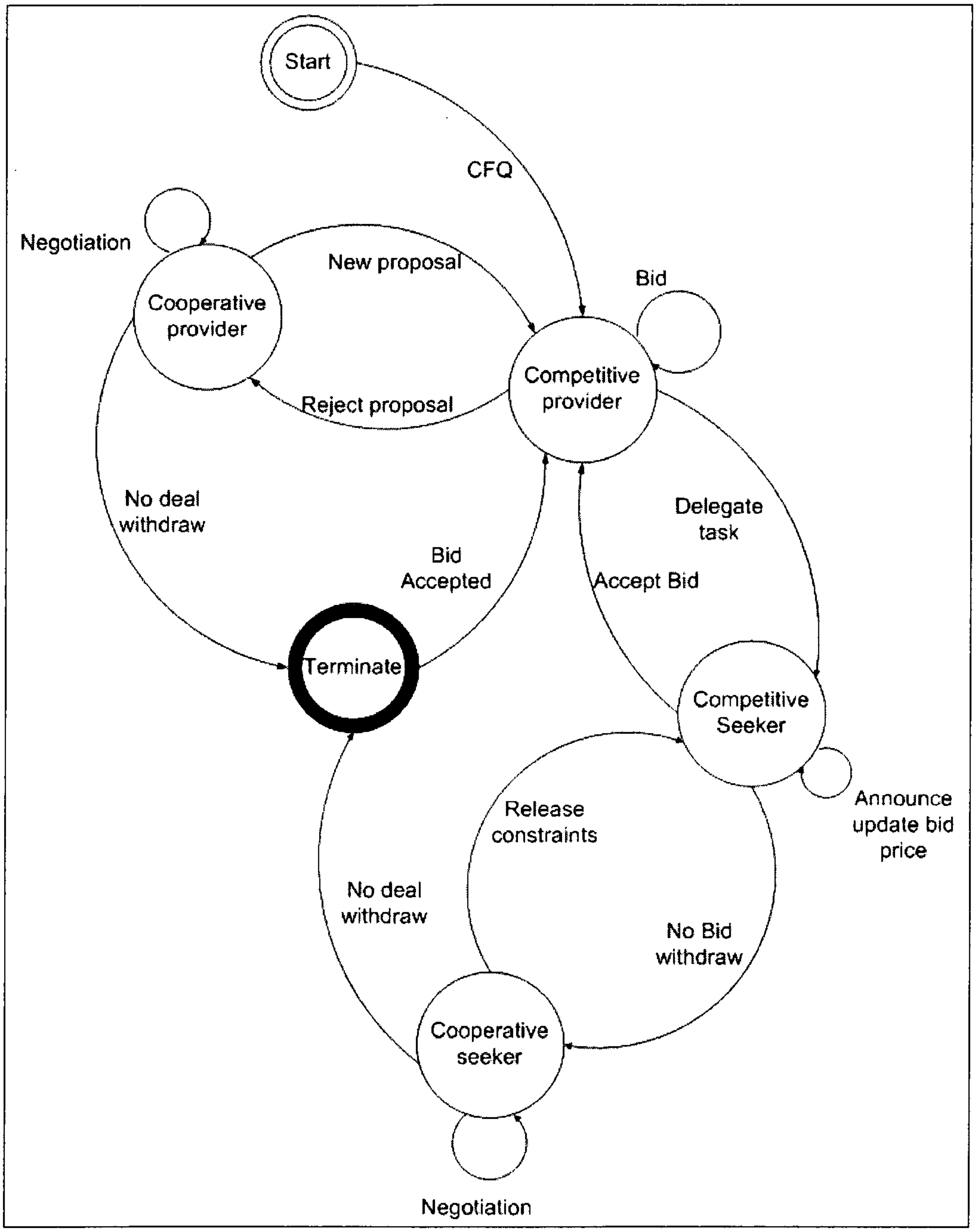


Figure 4: State transition graph

is in the *competitive seeker* state and calling for quote CFQ from another agent. Once the interested provider agent receives CFQ, the *competitive provider* state is triggered and agent makes a bidding. If it found that it needs to delegate some task, the *competitive seeker* state is triggered and in turn it announces a CFQ. Once an intermediate agent receives a successful offer from its subsequent agents and found it feasible, in turn it bids for the task to its precedent agent. If it does not found offer feasible, the *cooperative provider* state is triggered, and it rejects the CFQ with an argument supporting it's position. Also if its announcement for CFQ is refused by its subsequent agent accompanied by an argument of its position, the *cooperative seeker* state is triggered and it engages with its subsequent agent in negotiation cycle in order to reach a feasible deal. If a new proposal is agreed upon, a new cycle of bid is repeated to find a better solution (proposal) while at least one solution is guaranteed by agent who offer counter-proposal. This process iterates until one of the agent "accept" or "withdraw". Interaction terminate either by successful bidding or withdrawal as a result of failed negotiation, within the time limits.

3.4 Summary

The main problem in supply network formation is that existing interaction mechanisms employed may not be suitable to the agent's prevailing circumstances. This inflexibility means that the performance of both individual agents and the overall system may be compromised. To rectify this situation, a suitable interaction framework is required to address current challenges and issues in agent-based supply network coordination. Our approach is to design our adaptive interaction protocol by modeling what typically happens in the physical world of supply network. Adaptive interaction protocol is using a mix of coordination approaches to take advantage of each approach (contracting, bidding and negotiation). Adaptive interaction protocol is an agent behavior driven interaction protocol, as it adapts to agent behavior changes in response to changes in the environment. In order to achieve adherence between bidding and negotiation, contracting, a weighted multi-issue utility or payoff function is used to be the subject of contracting, bidding as well as negotiation.

Chapter 4

EXPERIMENTATION AND THEORY TESTING

" No society can surely be flourishing and happy, of which the far greater part of the members are poor and miserable." Adam Smith [116].

In this chapter, we justify our choice of MAS-based modeling section and simulation in Section 4.1. In Section 4.2, we describe an example of such implementation by presenting a simulation prototype of a case study used to validate the interaction protocol we propose. In Section 4.3 we provide analysis of the results. Finally in Section 4.4 we discuss the relative work of different modeling and simulation approaches.

4.1 Motivation for MAS-based Modeling and Simulation

In order to study the behavior of such a complex system, it is necessary to have a new model rather than traditional models. MAS-based modeling and simulation is a new modeling paradigm in modeling systems comprised of interacting autonomous agents. This paradigm offers a new way to elaborate systems that are decentralized rather than centralized, emergent rather than planned, and have concurrent rather than sequential control.

MAS-based modeling and simulation seems to be the most promising candidate for the development of the SN, since it extends significantly the capabilities of simulation in order to cope with new form of SN. An SN system can naturally be viewed as a society of MAS-based on the following reasons:

- The world is becoming increasingly challenging because systems that need to be managed are becoming complex. Many trends are contributing to increase in complexity, including shrinking in resources, growing structural complications, globalization, reduction in inventory, rising outsourcing, deepening information technology and expanding horizontal and vertical integration. Each of these trends increases the range of possible outcomes that must be considered by decision-makers.
- Both MAS and SN are composed of entities, which interact according to their roles and abilities. These entities have conflicting and competing individual requirements.
- MAS allows the automation of sharing and collecting supply network knowledge and information in the right time, which is essential to better decision making.
- Supply network is a special form of organization that can be regarded as a hybrid between a market relationship and a hierarchical organization. Therefore main characteristic of a supply network entities is cooperation and competition behavior, and as such it requires specific tools such as MAS.
- MAS-based modeling and simulation integrates different simulation models into a single environment without the need to adopt a common platform and language and to re-write. In addition large simulation can be split into smaller models in order to shorten simulation time.
- SN is constantly emerging depending on environmental conditions. With MAS, it is easy to follow the evolution of the SN by adding or removing agents

without the need of a complete reconstruction of the model since agents are autonomous entities.

- Distributed problem solving mechanism in multi-agent paradigm is consistent with SN's distributed decision making paradigm, since each entity behaves independently and autonomously, and pursues individual goals of achieving local optima, while satisfying both local and external constraints.
- MAS is suitable mean of representing the partners of SN, by delegating the agents to conduct the negotiation on behalf of the partners. The partners could then have the time to do the actual work required in the current SN.
- MAS captures and mimics the organization structure of a SN and the associated interacting mechanism, as each subsystem can be represented by an autonomous agent, which may itself be a multi-agent system containing other agents.

4.2 Design of Simulation Experiment Model

MAS-based modeling and simulation starts with simple rules of interaction among the individual components that drive the system to the complex behavior. It works bottom-up by examining what low-level rules and what kind of heterogeneous, autonomous agents are required to synthesize the required higher level behavior. For more work on MAS-based SN Simulation, see [6, 38, 58, 82, 92, 104, 119].

4.2.1 Assumptions Regarding MAS as SN Simulation Model

Since we focus on evaluating the effect of our interaction protocol on SN performance at the strategic level, we need to introduce assumptions that are necessary to simplify and abstract key processes.

- Each business entity in the network of companies will be served by a set of agents. Following the basic principle of division of work, each agent will perform a related set of functions, i.e., each agent will play a role in the node where it belongs. Each of these entities will form a "node" in a community of agents comprised of several entities in the network. The functionality across the SN is achieved through the interaction of the different nodes.

- In multi-agent encounter, negotiation is done between a pair of agents at a time.
- Each agent is a *utility maximizer*. In other word, each agent is rational and wants to maximize his own utility. Therefore agents have self-interest in looking for opportunities maximizing their benefits.
- Agents are honest about the importance of their goals and their options about how to achieve them.
- No agent has a view of the SN or is in a position to optimize the system as a whole. No single agent has complete information about the existing commitments of all agents in the SN organization, it has only information about subsequent agent to which he delegates sub-tasks.
- Agents have no memory and they are not able to learn. There is no consideration given by agents to the past. Each interaction (run) stands alone.
- All agents are using the same strategy, so the winner agent always is the agent who offers the lowest cost.
- Agent's Beliefs are constant, therefore agent's issues preferences (weights) is assumed static in simulation.
- All agents from the same market have symmetric abilities: they are able to perform the same set of actions in the environment, but with different efficiencies.
- Given a set of goals and constraints, agents search for a solution to optimize the goals and satisfy the constraints. Agents are constraint-based problem solver, given a set of goals and constraints, they search for a solution that achieves the goals and satisfies the constraints. They can generate more than one solution,

which enables the trade-offs by a set of cooperative agents.

- Agents have the ability and authority to relax some subset of constraints, within the global constraint.
- Self-interest agents can make effective decisions with local information, without knowing the private information and strategies of other agents.
- SN is a make to order pull system: no inventory or back order are assumed. The applied strategy is "Lot for Lot", where capacity is changed to adapt to orders fluctuations. Under this policy, the provided goods/services is exactly to the one ordered. Logistic costs such as inventory and transportation cost are out of cope of this thesis.
- In terms of the contractual setting, the supplier is assumed to charge a simple fixed unit price (no quantity discount scale).
- All partners in SN communicate only to its direct neighbors.
- Standard auction theory assumes that agents know their own utility for an item (private value, and maintain upper or lower bounds).
- Each agent is supposed to have a utility function in terms of three critical factors: price, time, and quality. The time and quality are predefined according to coordination proposed contract. The time value is the lead time of a resource. quality is function of standard deviation σ and process capability function. Finally the price is based on the customer's acceptance.
- The system is assumed to proceed in series of rounds where at every round an agent may receive task to process. A task request arrives randomly at any instant. No agent knows in advance what task it will receive or when.

4.2.2 Implementation Environment

Our system is implemented in Jadex [94]. It is based on the Java Agent Development framework (JADE) agent platform. Jadex used both XML and Java and is FIPA compliant agent environment. The framework consists of Application Program Interface (API), execution model, and a predefined reusable generic functionality. In order to develop an agent application in Jadex, one has to create two types of files: Java-based classes for plans implementation, and an XML-based Agent Definition File (ADF). Java-based classes for plans describe the actions that an agent undertakes. ADF specifies beliefs, goals, events, and plans for an agent. A configuration tag specifies initial beliefs, initial goals, initial events, and initial plans. In Jadex, the ADF is loaded first in order to start an agent, then the agent is initialized with beliefs, goals, and plans defined by configuration tag. Jadex platform architecture includes the following elements:

Container, which contains several agents. A specific container, the *Main container* contains a set of active containers. Other containers register with the *Main container* as soon as they start.

Agent management system, which provides the naming and control access service, like a white page, and represents the authority in the platform.

Directory facilitator, which provides a yellow pages services.

Jadex Agent Architecture

The reason behind using Jadex in our implementation is that it it fully compatible with BDI model and provides a BDI reasoning engine. Jadex agent architecture includes:

Belief-base: The belief-base is a set of agent's belief facts that make up the agent's knowledge. The belief-base contains strings such as the name of a belief that represents and identifier for a specific belief. Jadex uses Object Query Language (OQL) to search the condition that triggers plans or goals when some beliefs change. Beliefs are stored as expressions and evaluated dynamically on demand.

Goals: Goals are concrete, momentary desires of an agent. Unlike traditional BDI system, Jadex treats goals as events. In Jadex, an agent for any goal will engage into certain suitable actions. Some goals may only be valid in specific context determined by the agent's belief. When the context of goal is invalid, it will be suspended until the context is valid again. An ADF will include the contents of an agent's goal that represents the agent's desire. The agent exhibits different behavior according to the type of goal. There are four types of goals: *Achieve, maintain, perform and query.* An *achieve goal* just defines a desired target state, without specifying how to reach it. A *maintain goal* specifies a state that should be maintained once it is achieved. A *perform goal* states that something should be done but may not necessarily lead to any specific results. A *query goal* represents a need for information.

Plans: Plans describe the concrete actions that an agent may carry out to reach its goals. In Jadex, plans consist of two parts: plan head and plan body. The head contains the conditions under which the plan body is to be instantiated. These conditions are to be declared in the ADF. The plan body, written in Java-based classes, is the procedural set of steps describing the actions to achieve a goal or react to some event. Plans are selected in response to occurring events or goals, and are selected automatically by the system.

Events: In Jadex, these events are presented in the ADF. There are two types of events; message events and internal events. Internal events can be used to denote an occurrence inside an agent, while message event represent a communication between two or more agents. Events are usually handled by plans.

4.2.3 Supply Network Conceptual Model

In this work, we focus on the design and efficiency of coordination mechanism, by having each agent run a very simple algorithm. In order to study interaction model, we reformulate SN formation as coordinated distributed task allocation problem solving under dynamic environment, where SN entities relationships are viewed as customer-supplier pairs who interact with another to reach a mutually agreeable plans. The purpose is to demonstrate the applicability of the adaptive interaction protocol to this framework. This simulation would include the following building blocks:

Markets

Our simulation model represents a three echelon network system involving interaction associated between suppliers, intermediaries, and initiator of tasks. Multiple agents of certain type exist at each echelon forming a supply network organization (Figure 5).

- Customer market, which contains one customer asking for certain number of products;
- Distributer (intermediary) market, which buys the products from the supplier and adds up its margin, (there are three distributers) one distributer in each market set, each distributer is served by one supplier market;
- Supplier market, provides the products for distributers. We have three supplier markets, and at each market there are three agents.

Product

There is only one commodity, the product contains one sub-assembly part. An assembly is required by intermediaries according to bill for materials.

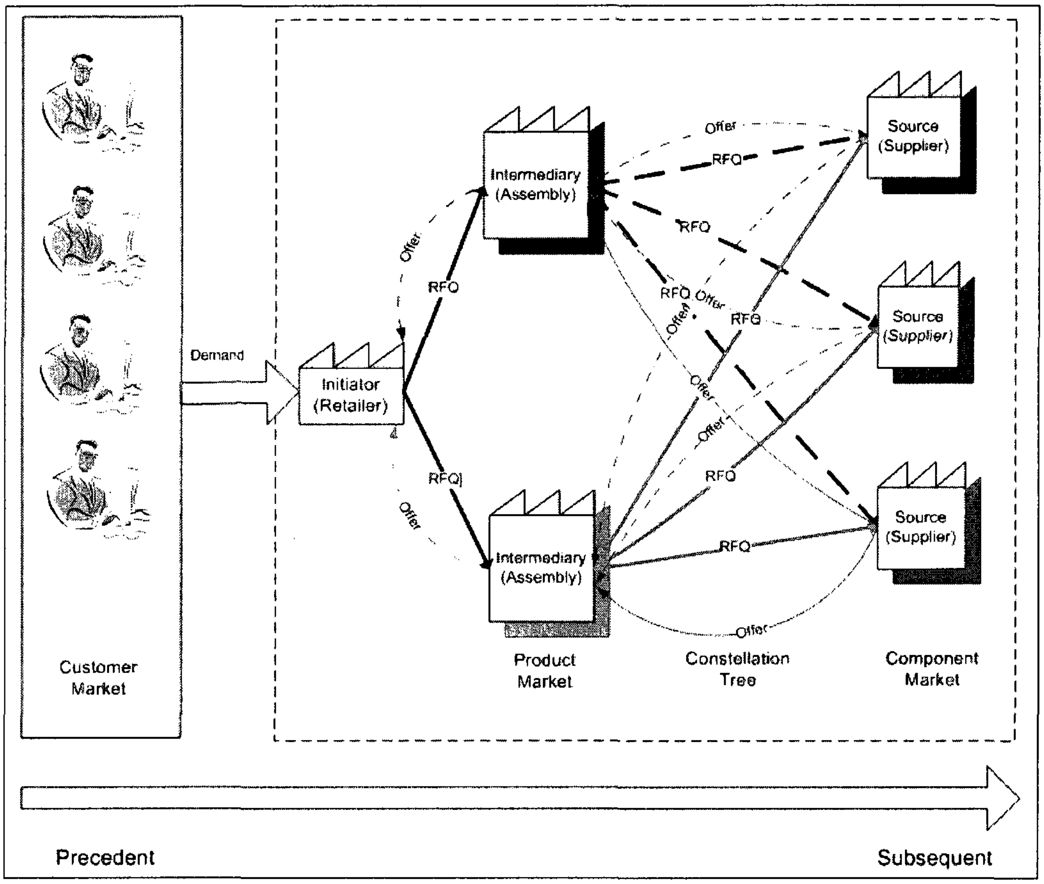


Figure 5: Supply network

Contract proposal issues:

Contract proposal contains order information such as lead time, product quality and quantity.

Quality The product/service must have quality equal to or better than customer specifications. The main condition for quality agreement is $\sigma_D \geq \sigma_C$ and $\sigma_S \geq \sigma_D$, where σ_D is distributer quality, σ_C is customer quality and σ_S is supplier quality.

Time Delivery or lead time is the length of time needed to deliver a product from the order to receipt by the customer. Total lead time $t_L(N_i)$ is calculated as follows: $t_L(N_i) = \frac{Q_O}{R_P} \times \{hrs/week\} + t_s + t_L(N_{i+1})$, where t_L is the lead time of node (N_i), Q_O is order quantity, and R_P is process capacity rate $unit/hrs$, t_s is the setup time, and $t_L(N_{i+1})$ is the lead time of the subsequent node. Lead time for tasks delegated is decided as $t_L(T_i) = max(0, t_L(ag_i) - t_L(ag_{i+1}))$, where T_i is the task to be delegated by agent ag_i, $t_L(ag_i)$ is the maximum lead time required by agent ag_i, and $t_L(ag_{i+1})$ is the lead time of agent ag_{i+1} in the subsequent node according to his capacity and preceding nodes constraints. Lead time functions has additive property, so the the total lead time is simply the sum of lead times from source node to the current node. Lead time function parameters $< R_P, t_s >$ are stochastic values, determined by continues random number generator. The main condition for lead time agreement is $t_{L_C} \geq t_{L_D} + t_{L_S}$, where t_{L_C} is the customer required lead time, t_{L_D} is the distributer lead time and t_{L_S} is the supplier lead time.

Cost Total Cost C_T is calculated as follows: $C_T = Q \times C_V + C_F$ where C_V represent variable cost/unit, and C_F represents fixed cost. Fixed cost includes financial cost and setup time cost etc., while variable cost includes products/services provided by subsequent node such as cost of material, energy etc. Cost function is additive so that every cost comes from subsequent stages are simply added to current node cost. Cost function parameters $< C_V, C_F >$ stochastic values are determined at each run using discrete random number generator. The main condition for cost agreement is $P_{C,max} \geq C_{D,min}$ and $P_{D,max} \geq C_{S,min}$, where $P_{C,max}$ is the maximum price a customer is willing to pay, $C_{D,min}$ is the distributer minimum cost, $P_{D,max}$ is the maximum price that the distributer is willing to pay for supplies and $C_{S,min}$ is the minimum supplier cost.

4.2.4 Simulation Setup

Simulation Initialization

Initial values are randomly generated for reserved price limits for customer, distributer and supplier. Quality and lead time are also randomly generated. In order to create randomness, we use standard normal distribution function that is built in "Java". The notation

Variable	Value
α	0.3
β	0.3
γ	0.3
σ_C	min(6, Rnd[0,1]*5+2)
t_{L_C}	max(Rnd[0,1]*10+4, Rnd[0,1]*10)
$P_{\{C,max\}}$	Rnd[0,1]*40+60

Table 2: Customer attributes random variables

$N(\mu, \sigma^2)$ indicates that the random variable X is normally distributed with mean $\mu = 0$ and variance $\sigma^2 = 1$. By generating random initial values for every run we can create different simulation scenarios. Simulation will terminate according to the mentioned conditions in (4.2.4). In order to generate statistically independent and identically distributed replications, simulation replication is repeated for 30 times each run, and after each replication, initial values are cleared. Since different runs use different random numbers and the same initialization value, their comparable random variables from the different runs are independent and identically distributed [4].

Simulation agents

Customer attributes' random variables Customer utility function is as follows:

$$Utility_C(\delta) = \alpha u(1/p) + \beta u(\frac{1}{t_L}) + \gamma u(\sigma_C) \tag{11}$$

Where α, β, γ are preference weight, $u(1/p)$ is the utility function of price, $u(\frac{1}{t_L})$ is the utility function of lead time and $u(\sigma_C)$ is the utility function of quality (see Table 2).

Distributor attributes' random variables Distributor payoff function (Definition 3.1.2) is as follows:

$$Payoff_D(\delta) = Reward(\delta) - Cost_D(\delta) \tag{12}$$

Variable	Value
$C_{\{D,min\}}$	supplier cost*1.1
t_{L_D}	Supplier lead time
σ_D	Customer required quality

Table 3: Distributer attributes random variables

Variable	Mean
$C_{\{S,min\}}$	((Rnd[0,1]*40+60)*1.1)
t_{L_S}	max(Rnd[0,1]*10+4, Rnd[0,1]*10)
σ_S	min(6, Rnd[0,1]*5+2)

Table 4: Supplier attributes random variables

where *Reward* is current distribute bidding price, $Cost_D(\delta) = (\sum_1^k(P_{S,k}) + C_D)$ is the total cost of the deal δ, k is the number of sub-task delegated, C_D the distributing cost and $P_{S,k}$ is the price of each sub-assembly (see Table 3).

Distributors get the bidding prices from suppliers and multiply their price by their margin (e.g., $1.2 \times Cost$) in order to decide the selling price. As for time and quality, they abide by customer and supplier constraints.

Supplier attributes' random variables Supplier payoff function is as follows:

$$Payoff_S(\delta) = Reward(\delta) - Cost_S(\delta) \tag{13}$$

where *Reward* is the current supplier bidding price, $Cost_S$ is the cost of the raw materials (see Table 4).

Auction parameters

Auction ending time: The bidding ending time is determined by the auctioneer agent. Intermediary agents in turn, announce what is left of bidding time to the subsequent bidders.

Bid elapsed interval time: The time interval agent should bid within before announcing the winner, it is to be determined by auctioneer agent.

Bid decrement: The amount by which a bid is decreased each time the current bid is outdone. Biddingdecrement size is decided during runtime by agent using (Equation 6) as discussed in Chapter three.

Payoff-utility-based negotiation

For every seeker there is a reserved payoff/utility value for its payoff/utility function (e.g., Equation 11), which is the sum of utilities of each attribute (in case of using utility only). The winning proposal is the proposal who offers seeker the highest payoff/utility. If providers' constraints restricts the global constraints, seeker reserved issues values may changed according to SN providers constraints' values. The seeker change its constraints, if it gets equal or higher payoff/utility value from new proposals.

Simulation stopping criterion

The main stopping criteria is:

- Success in SN formulation;
- Failure of negotiation;
- Expiry of Auction designated time.

4.3 Simulation Experiment Results

In order to evaluate the effectiveness of our protocol, a number of experiments has to be carried. These experiments serve to quantify its ability to reduce the number failures. The behavior of the implemented system is studied by randomly varying the task load, as well as agent's capabilities across the MAS. In order to test the protocol, we have to compare the system under two different scenario: (1) using bidding only and (2) using

	No of successful replications	Distributer wining Price	Supplier winning price	Customer Utilities Value
Average	17	88	73	1.40
Standard deviation		45	37	0.77

Table 5: Simulation summary results without negotiation

	No of successful replications	Distributer wining Price	Supplier winning price	Customer Utilities Value
Average	27	93	79	1.55
Standard deviation		31	26	0.59

Table 6: Simulation summary results with negotiation

bidding and negotiation. The outputs of both scenarios then is evaluated in terms of some criteria (Subsection 4.3.1). The simulation is repeated a total of 60 replications and each scenario runs for 30 replications. First simulation scenario (bidding only) output data are shown in (Table 7) in (Appendix A) . Second simulation scenario (bidding and negotiation) output data are shown in (Table 8) in (Appendix B).

4.3.1 Analysis of Results

We based our evaluation criteria on Bidding only simulation summary results as shown in (Table 5), and on Bidding and negotiation simulation summary results as shown in Table (6).

Evaluation Criteria

Customer satisfaction Our protocol raises average customer utility gained, which is a measure of customer satisfaction, from 1.40 to 1.55 as shown in (Figure 6).

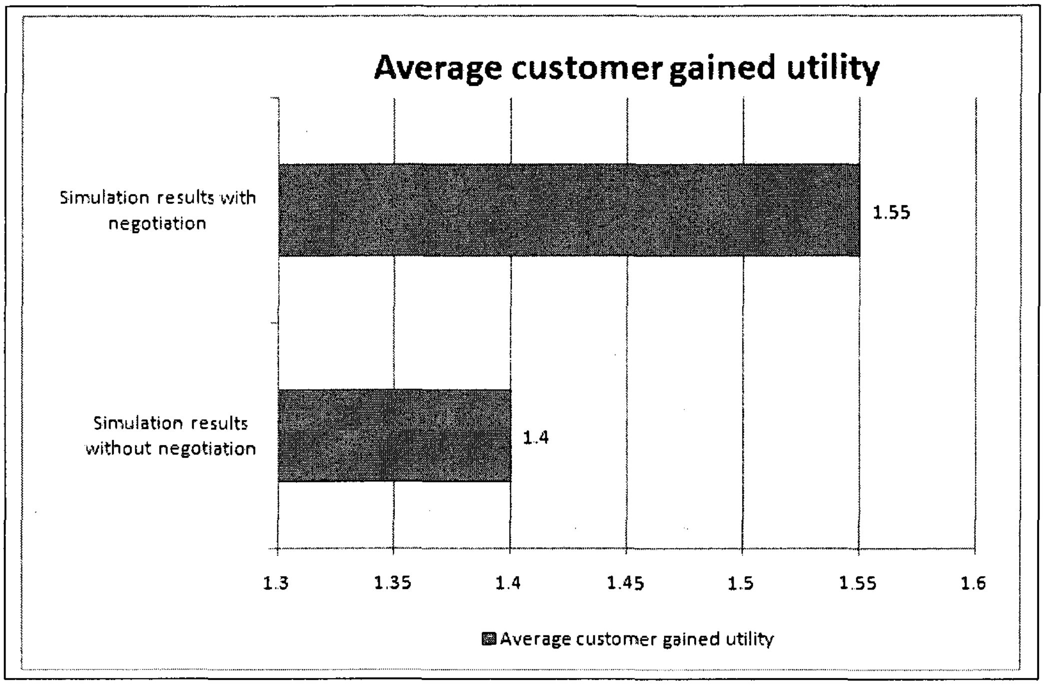


Figure 6: Average customer gained utility

SN formation success A protocol supports SN formation success, if it eventually ensures that agreement is certain to be reached (number of successful SN formations per total replications). In the total of 13 replications the bid fails, the utility-based negotiation is triggered and the number of successful replications raises from 17 to only 27. The number of failed runs drops from 13 replications to 3 replications as illustrated in (Figure 7).

Quality of solution We use *social welfare* [24, 81] as an indicator of solution quality. We distinguish two types of criteria when assessing the quality of a task allocation namely *collectiveness* and *fairness*, both can be described in terms of winning price or utility function.

The utilitarian social welfare of an allocation is the sum of individual's payoff or utility experienced by the society members in a given solution $\mathrm{SW}_u(os_i) = \sum_{oe_j \in os_i} \mathrm{payoff}(oe_j)$. It measures the global good of the agents. By comparing results from both scenarios, average customer utility function per 30 runs in utility-based negotiation is 1.55, which is higher

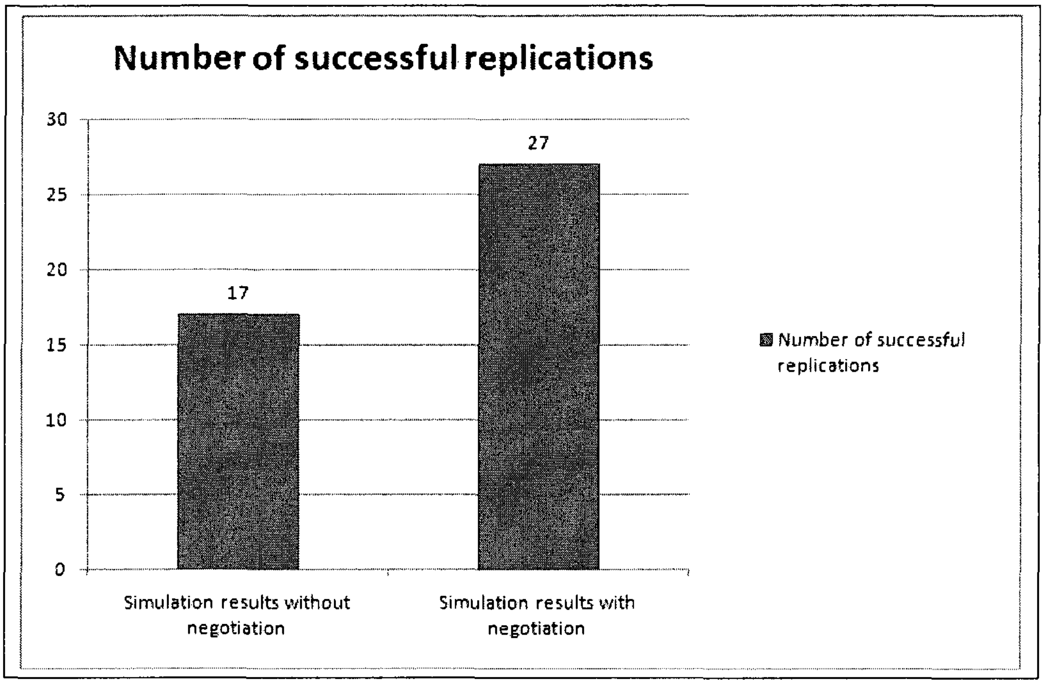


Figure 7: Number of successful replications

than average customer utility function results from bidding only, which is 1.40. At the same time, distributer winning price moves up from 88 to 93 and the supplier winning price moves up from 73 to 79. This indicates that our protocol maximizes social welfare by maximizing the sum of the utilities of participants (see Figures 8 and 6).

The egalitarian social welfare of an allocation is a measurement of fairness among SN agents, which is the variance of the individual utilities gained and winning prices. The standard deviation of both the winning prices and utility gained are measures of fairness of welfare $\mathrm{SW}_e(os_i) = \underset{oe_j \in os_i}{std.dev.}\, \mathrm{payoff}(oe_j)$. The customer's utility standard deviation drops down from 0.77 in bidding scenario to 0.59 in case of bidding and negotiation scenario. Same same applies to winning prices, for distributers winning prices standard deviation drops down from 45 to 31 and the suppliers winning price standard deviation drops down from 37 to 26, (see Figures 9 and 10).

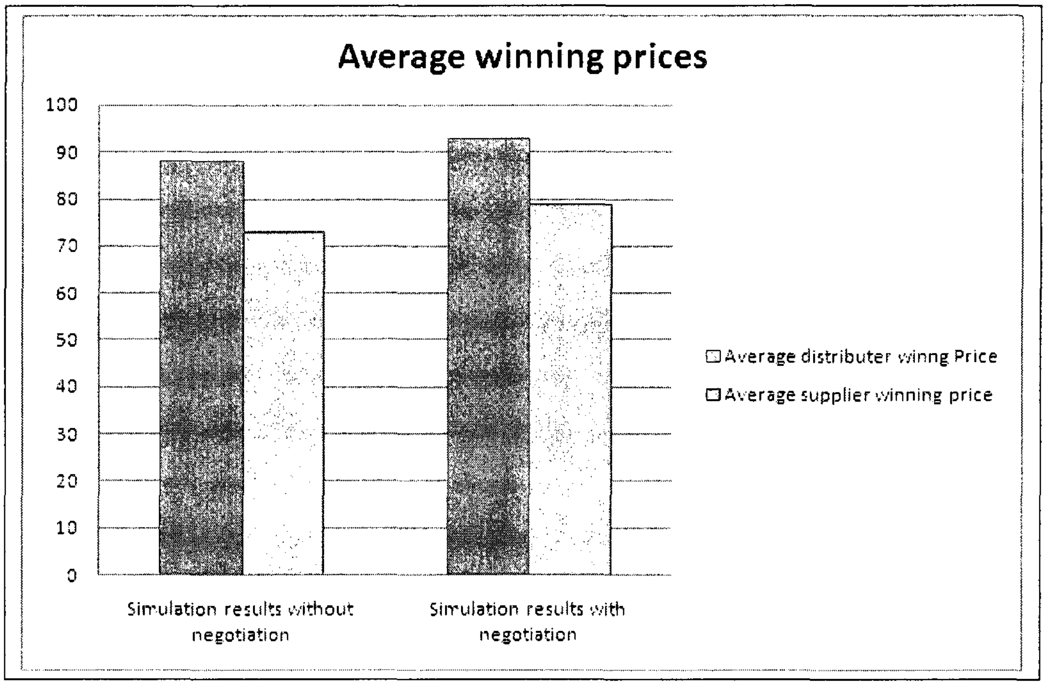


Figure 8: Average winning prices

4.4 Related Modeling and Simulation Work

SNs are complex systems in terms of number of entities involved, activities and decision variables. Complexity comes from the potential number of interactions, occurrence of unexpected events, and lack of details during the planning process. Therefore, SN optimization presents a great challenge.

Since each SN has its own unique and complex features, it is necessary to abstract reality and use a simplified copy which is a model, and to have behavior analysis tools available to support decision making.

4.4.1 Organizational Modeling Approach

SCOR

The Supply Chain Operations Reference model (SCOR) [108] is developed by the Supply Chain Council (SCC). The SCOR-model is a top-down model that is based on three

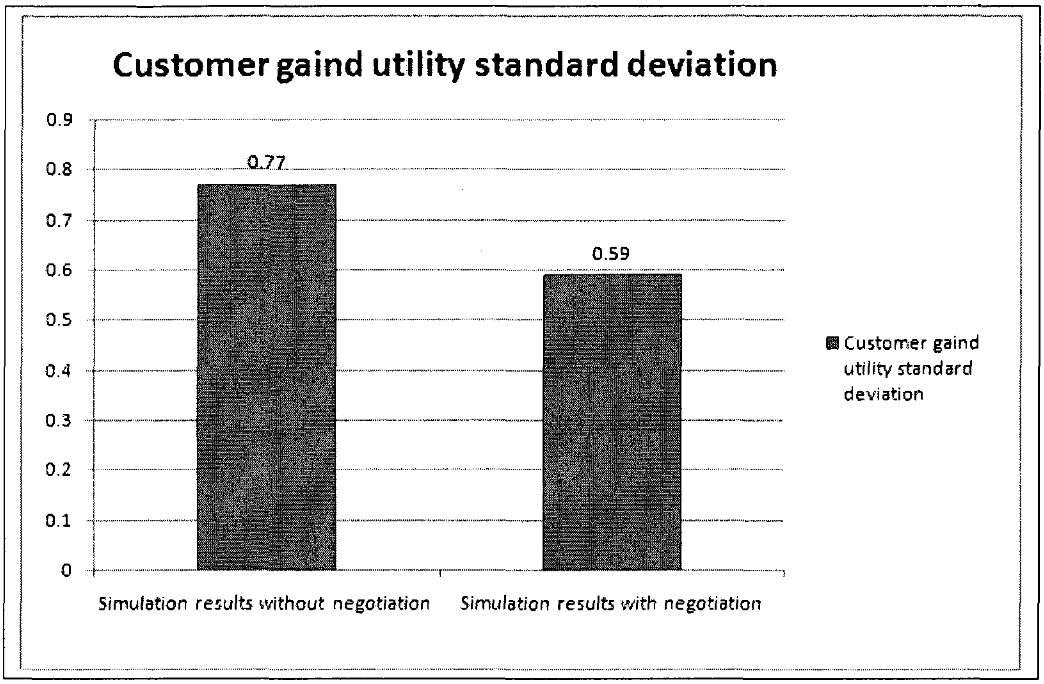


Figure 9: Customer gained utility standard deviation

different methodologies: Business Process Re-engineering (BPR), benchmarks, and analysis of best practice. The SCOR model is used to translate business strategy into a supply network architecture designed to achieve specific business objectives. The SCOR-model is structured around five supply network processes: Plan, Source, Make, Deliver, and Return. SCOR process design is defined by configuration of materials, work and information flow with SCOR's plan. The SCOR model has four levels of detail, the first three are process, sub-processes and activities are described in the model, the fourth level, which is detailed work-flow level tasks is always customized to an organization's specific strategy and requirements. The SCOR model lacks integration between supply network processes, moreover it is only a reference model and not an optimizing model since it has no formal mathematical description of supply network. For application of SCOR in SN, see [7, 39].

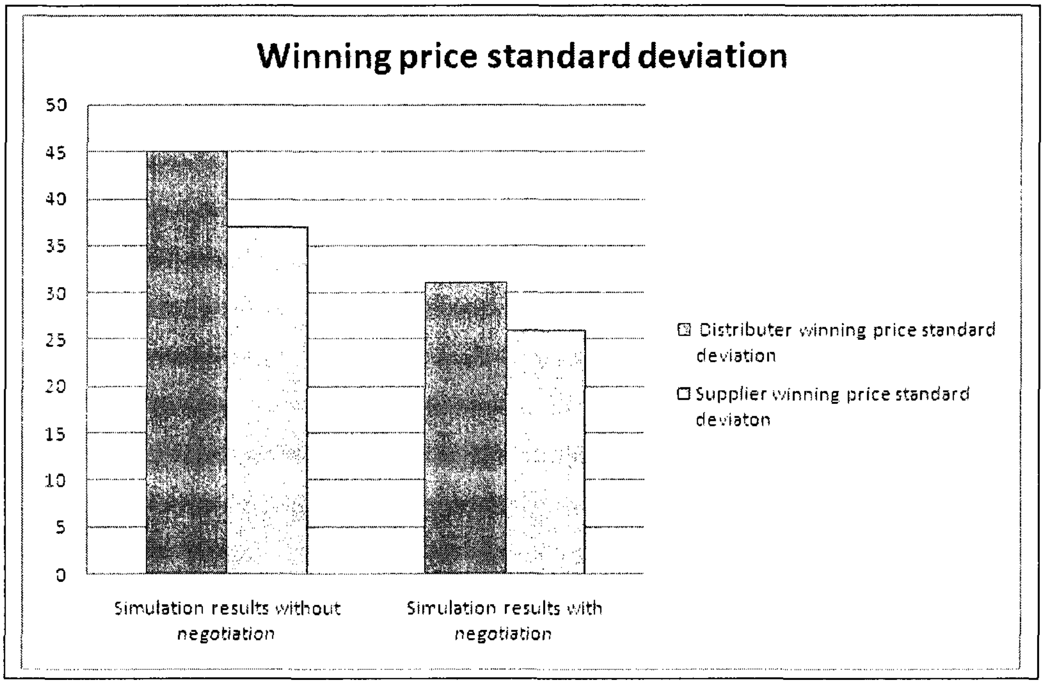


Figure 10: Winning prices standard deviation

CPFR

Collaborative Planning Forecasting and Replenishment (CPFR) [129] developed by the Voluntary Interindustry Commerce Standards (VICS) offers guidelines on developing shared processes among the collaborating partners for joint planning. This abbreviation is used to identify a nine-step approach, which provides volunteer standards, protocols, guidelines, etc., which are required or exchanging sales and order forecasts (on a web-based platform) between trading partners (conventionally identified as the buyer and seller) belonging to the same supply network. Under CPFR, both buyer and seller collaborate by correcting, adjusting, proposing prices and quantities to reach an agreement on a unique forecast, so that the buyer's purchases forecast and the seller's sales forecast coincide. However, drawbacks of CPFR exist, and include trust issues (companies are reluctant to share the level of data due to a potential loss of competitive information), technology capability issues, and high implementation costs. For application of CPFR in SN, see [17, 18].

Despite its drawbacks organizational models are basis for supply network ontology modeling since they are widely shared supply network concepts [70, 76, 83].

4.4.2 System Dynamics Modeling Approach

This approach relies on control theory. The SN studies that are based on control theory make an analogy between SN and control systems. In this approach ordinary differential equations (ODE) are used to model SN. The model structure consists of multiple interacting feedback loops as basic building blocks of the methodology. These feedback loop represents the policies. The main advantage of using control theory is the possibility of deducing the occurrence of a specific behavior mode because of a certain structure, while the main drawback of using traditional control theory for supply network is that the structure has to be predetermined before starting the simulation. For application of system dynamics in SN, see [37, 121, 122, 123].

4.4.3 Operations Research Modeling Approach

This group includes mainly approaches that use classic operations research, e.g., optimization theory and statistical methods to solve real-life problems in operations optimization such as plant design, production scheduling, logistics of distribution and inventory management. However, this approach also has some limitations [9, 57]: it is technically insufficient in handling a high volume of what-if scenarios, and it is very difficult to solve a problem where more than two issues are considered. For more details about work on SN formation using operations research, see [27, 66, 124, 130, 134].

4.4.4 Economic Modeling Approach

Game Theory (GT) is a branch of mathematics concerned with decision-making in social interactions. GT can be defined as the study of mathematical models of conflict and cooperation between intelligent rational decision makers. It applies to situations where there are two or more players each attempting to choose between two strategies. The possible outcomes of a game depend on the choices made by all players, and can be ranked in order of preference by each player. GT provides general mathematical techniques for analyzing situations in which two or more players make decisions that will influence one another's welfare.

A game in the normal form consists of: players, a set of strategies, and payoffs received by each player. With the invention of the Internet and the reduction of information technology cost, certain game theory tools such as auction theory, contract theory, bargaining theory, and coalition theory have received attention. GT based approaches assume that players are utility maximizers. It deals with encounter of fully rational and self-motivated players in an open system. A game is a form of "strategic encounter" between a number of players. Each player has a set of alternative actions (strategies) available to it. A central concept in GT is *Nash equilibrium*, where no player has an incentive to deviate from a particular strategy, given that other players stick to their strategies. GT is an essential tool in analyzing SN entities that have conflicting objectives, and where its decisions affect each other's payoff. As such GT deals with interactive optimization problems. The participants engage in negotiation according to their respective bargaining strategies to arrive at a "fair" price for the item. Each player's strategy guides its actions at various steps in the game is based on the available information. For specific applications of GT to SN, see [13, 14, 29, 80, 125].

On the downside, GT is *normative* theory (it tells us what a partner should do), but it does not show how to implement such decisions efficiently. In addition, GT is *quantitative* in nature. Third, GT is unable to deal with increased complexity and uncertainty inherent in many real-life situation. Finally, the design of effective economic mechanism depends on the model used to describe the interaction between participants.

Chapter 5

CONCLUSIONS AND FUTURE WORK

" The rich ... divide with the poor the produce of all their improvements. They are led by an invisible hand to make nearly the same distribution of the necessaries of life which would have been made, had the earth been divided into equal proportions among all its inhabitants.." Adam Smith [117].

In this chapter we present summary of the thesis in Section 5.1. In Section 5.2 we furnish again our contributions. In Section 5.3 we describe the limitation of this work. Finally in Section 5.4 we propose the future work of this thesis.

5.1 Summary

Good design is the key determinant of effectiveness and efficiency of the supply network. While quantitative model provide solutions and decision support for the management in different supply network subsystems, the most challenging problem is to develop an efficient modeling and analyzing technique for supply network integration and coordination problem in order to gain understanding of performance. Research generally concentrates on smaller parts of the system without the consideration of integration and subsystems interdependency. Therefore, it is imperative to employ both analytical and simulation-based techniques, in conjunction with existing methods to develop models for integrated and coordinated supply network problems.

This dissertation has addressed supply network formation problem areas. We have argued that autonomous agents provide an effective solution but need to be given the flexibility to dynamically select the mechanism they use for coordinating their activities during cooperative distributed problem solving. We have presented an adaptive interaction protocol framework that enables agents to dynamically select the most appropriate coordination mechanism in a given situation.

When the environment in which decision making takes place is dynamic, open, uncertain, and heterogeneous, making decisions about coordination is difficult. This is because it is impossible to enumerate in advance the wide variety of contexts in which coordination is likely to be needed. To this end, we have developed a decision making framework that allows agents to choose their coordination mechanism to the prevailing context. Our means to achieve the thesis objective is through using multi-agent systems as a simulation model for bilateral interaction protocol for supply network formation.

Multi-agent systems are used as a mean for saving costs and resources. MAS are loosely coupled networks of software agents that interact to solve problems that are beyond individual capacities or knowledge of each problem solver. The general goal of interaction protocol is to create agent-based SN system that interconnects separately developed agents.

This dissertation gave an overview of supply network coordination mechanism in general and a taxonomy of supply network modeling. A classification of recent research work within three groups was also given. These groups are: (1) contract-based coordination; (2) auction and bidding-based coordination; and (3) negotiation-based coordination. Second, the thesis proposed to combine contracting, bidding, and negotiation in a novel way in order to create even more efficient interaction protocol for supply network formation. A new interaction protocol based upon agent technology has been defined to verify the logical property of supply chain interaction. The third part presented multi-agent-based simulation system for the new proposed interaction protocol. The final part of this dissertation used agent-based simulation concepts to evaluate the success of three-echelon supply network and investigate the impact of different scenarios on the proposed interaction approach.

Although our protocol is intended to be adaptive to all SN life cycle phases, we focused our simulation on the SN formation phase. The formation of SN is considered within the context of an electronic market place where several parties compete to become partners of SN. Each market place provides a meeting place and yellow pages for agents that are represented by their experiences, competencies and availability. The automatic match making and SN formation is supported by our protocol.

We showed that in our simulation scenario, agents are more successful when they have the ability to dynamically select the coordination mechanism during run-time. The simulated experiment was validated using a range of tests showing that dynamic selection of coordination mechanisms increases the overall performance. Our protocol is adaptable to agent behaviours from competitive bidders to cooperative negotiators in order to find the desired SN organizational structure.

5.2 Contributions

In this thesis, we have proposed and developed a bilateral solution to automated interaction protocol for supply network formation. To the best of our knowledge, the automated system we have outlined in this thesis is the first solution to automated bilateral interaction protocol using contracting, bidding, and negotiation. More specifically, the results and contributions of this thesis are as follows:

- A theoretical contribution in designing an interaction protocol that meets some basic desiderata for agent interaction protocols. This protocol is dynamic and adaptive to agent's changing behavior through the SN formation stages.

- The consideration of supply network life cycle within a framework of coordinated distributed problem solving approach.
- A framework for supply network simulation and modeling approach using multi-agent systems as a test bed.
- An operational prototype using Jadex BDI reasoning agents. The prototype serves as a proof of concept of the proposed bilateral interaction protocol.

5.3 Limitations

The proposed multi-issue utility-based negotiation might overcome the problem of empty negotiation space but on the account of complexity. The negotiation space grows as the number of attributes and the number of possible attributes values increase. When there is a time constraint, it is infeasible to consider every possible solution. Another major limitation is the data used was from theoretical distribution and was not from a real world system. A third limitation is despite benefits of multi-agent systems in distributed problem solving, optima cannot be guaranteed because there is no global view of the system, MAS are less efficient than centralized solution because systems and data are in independent organizations, and decision-making is based on local knowledge. A fourth limitation, multi-issue utility/payoff function is non-linear equation and may contains several local optima and the basic bid decrement iterative improvement may obtain only local optima, therefor it is not guaranteed to find global optima. Finally, prediction for autonomous agent can usually be made only at the aggregate level since agents' behavior cannot be explained in detail because most agents are constructed as a black-box system and/or their behavior is determined by "non-transparent" schema (e.g., by applying genetic algorithms, artificial neural networks, etc.).

5.4 Future Work

This thesis is about designing a bilateral interaction protocol using multi-agent-based negotiation system. It opens the door for several research opportunities. Further research may include:

- Implementing shared ontology in order to create more realistic situations.
- Extending negotiation to a more flexible argumentative negotiation. Argumentation provides means for structuring dialogues between participants that have potentially conflicting viewpoints. In particular, argumentation provides a framework for ensuring that interaction respects certain principles (e.g., consistency of each participants statements).
- Since we assume vertical integration only, our protocol can be extended to accommodate horizontal integration.
- Considering agents with non-homogeneous strategies including malicious agents and using mixed game theory to analyze the results.
- Applying artificial intelligence tools within an agent (e.g., neural network and machine learning) and studying the changes of its behavior on the basis of the experience gained.
- A combination of agent-based approach with semantic web technology and web services may introduce some benefits by offering interpretability among heterogeneous agents, facilitating use of ontologies and providing better semantic support in tasks descriptions.
- More investigation is needed in the area of determining the bid decrementation step size in order to balance between bidding speed and fining optimal solution. In addition, applying and comparing search techniques may find the more optimal techniques for finding a deal from the negotiation space.

- Future research should also focus on further validation of this study using techniques such as comparison with other models, historical data validation, and formal verification techniques such as proofs, model validation and verification etc.
- Finally, In order to determine its applicability to the real world, the computational methodology addressed in this thesis needs to be tested by real supply networks.

Bibliography

[1] L. Amgoud and N. Parsons, S.and Maudet. Arguments, dialogue, and negotiation. In W. Horn, editor, *European Conference on Artificial Intelligence (ECAI 2000)*, page 338342, 2000.

[2] James B. Ayers. *Handbook of Supply Chain Management.* St. Lucie Press, Boca Raton, 2001.

[3] Moshe Babaioff and Noam Nisan. Concurrent auctions across the supply chain. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 1–10, New York, NY, USA, 2001. ACM.

[4] Jerry Banks, John Carson, Barry L. Nelson, and David Nicol. *Discrete-Event System Simulation, Fourth Edition.* Prentice Hall, December 2004.

[5] K. S. Barber, A. Goel, C. E. Martin, K. Suzanne Barber, Anuj Goel, Cheryl, E. Martin, and Cemartin. Dynamic adaptive autonomy in multi-agent systems. *In Journal of Experimental and Theoretical Artificial Intelligence*, 12:129–147, 2000.

[6] M. Barbuceanu, R. Teigen, and M.S. Fox. Agent based design and simulation of supply chain systems. In *Proceedings of the 6th Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises (WET-ICE'97)*, pages 18–20, 1997.

[7] Michael W. Barnett and Charles J. Miller. Analysis of the virtual enterprise using distributed supply chain modeling and simulation: an application of e-scor. In *WSC*

'00: Proceedings of the 32nd conference on Winter simulation, pages 352–355, San Diego, CA, USA, 2000. Society for Computer Simulation International.

[8] J. Beck and M. Fox. Supply chain coordination via mediated constraint relaxation. In *Proc. 1st Canadian Workshop on Distributed Artificial Intelligence*, pages 61–72, Banff (Alberta, Canada), 1994.

[9] G. Berning, M. Brandenburg, K. Gürsoy, J. Kussi, V. Mehta, and F.-T. Tölle. An integrated system solution for supply chain optimization in the chemical process industry. *FOCAPO 2003. 4th Int. Conf. of Computer-Aided Process Operations, Proceedings of the Conference held at Coral Springs, Florida, January 12-15, 2003*, pages 45–58, 2003. See C03-2.

[10] Felix Brandt and Gerhard Wei. Vicious strategies for vickrey auctions. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 71–72, New York, NY, USA, 2001. ACM.

[11] Michael Bratman. *Intention,Plans and Practical Reason*. Harvard University Press, Cambridge, MA, USA., 1987.

[12] David A. Burke and Kenneth N. Brown. A constraint based agent for tac-scm. Technical report, Cork Constraint Computation Center, University College Cordk, 2005.

[13] G. Cachon and S. Netessine. *Game theory in supply chain analysis*. Kluwer, 2004.

[14] G. P. Cachon. Supply chain coordination with contracts. In S. Graves and T. de Kok, editors, *Handbooks in Operations Research and Management Science*. North Holland Press, 2004.

[15] Gérard P. Cachon and Martin A. Lariviere. Contracting to assure supply: How to share demand forecasts in a supply chain. *Management Science*, 47(5):629–646, 2001.

[16] G.P. Cachon. Supply chain coordiantion with contracts. 2001.

[17] M. Caridi, R. Cigolini, and D. De Marco. Linking autonomous agents to cpfr to improve scm. *journal of Enterprise Information Management*, 19(5):465–482, 2006.

[18] Maria Caridi and Sergio Cavalieri. Multi-agent systems in production planning and control: an overview. *Production Planning & Control*, 15(2):106–118, 2004.

[19] Kathleen M. Carley and Les Gasser. Computational organization theory. pages 299–330, 1999.

[20] Christiano Castelfranchi and Rino Falcone. Social trust: A cognitive approach. In Christiano Castelfranchi and Yao-Hua Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer Academic Publishers, 2001.

[21] Cristiano Castelfranchi and Rino Falcone. Towards a theory of delegation for agent-based systems. *Robotics and Autonomous Systems*, 24:141–157, 1998.

[22] Alexander Smirnov Charu, Er V. Smirnov, and Charu Ch. Ontology-based knowledge management for co-operative supply chain configuration. In *American Association for Artificial Intelligence (AAAI) Symposium*, pages 85–92. AAAI Press, 2000.

[23] Fangruo Chen, Awi Federgruen, and Yu-Sheng Zheng. Coordination mechanisms for a distribution system with one supplier and multiple retailers. *Manage. Sci.*, 47(5):693–708, 2001.

[24] John S. Chipman and James C. Moore. The new welfare economics 1939-1974. *International Economic Review*, 19(3):547–584, 1978.

[25] P. Chopra, S.and Meindl. *Supply Chain Management-Strategy, Planning, and Operation*. Upper Saddle River, 2001.

[26] M. Christopher. *Logistics and Supply Chain Management-Strategies for Reducing Cost and Improving Service*. Financial Times/Prentice Hall, London, 1998.

[27] Morris A. Cohen and Hau L. Lee. Strategic analysis of integrated production-distributed systems: models and methods. *Oper. Res.*, 36(2):216–228, 1988.

[28] Clyde H. Coombs and George S. Avrunin. *The Structure of Conflict*. Lawrence Erlbaum, 1988.

[29] C. J. Corbett and C. S. Tang. *Designing supply contracts: Contract type and information asymmetry*, pages 372–391. Kluwer Academic Publishers, Norwell, MA, 1999.

[30] Charles J. Corbett, Deming Zhou, and Christopher S. Tang. Designing supply contracts: Contract type and information asymmetry. *Management Science*, 50(4):550–559, 2004.

[31] Daniel David Corkill. *A framework for organizational self-design in distributed problem solving networks*. PhD thesis, University of Massachusetts Amherst, 1983.

[32] Edmund H. Durfee. Distributed problem solving and planning. pages 118–149, 2001.

[33] Ulle Endriss and Nicolas Maudet. Welfare engineering in multiagent systems. In *In Engineering Societies in the Agents World IV*, pages 93–106. Springer-Verlag, 2004.

[34] Cora B. Excelente-Toledo and Nicholas R. Jennings. Learning to select a coordination mechanism. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1106–1113, New York, NY, USA, 2002. ACM.

[35] J. Ferber. *Multi-Agent Systems*. Addison -Wesley, Reading,MA, 1999.

[36] Erastos Filos and Eoin Banahan. Towards the smart organization: An emerging organizational paradigm and the contribution of the european rtd programs. *Journal of Intelligent Manufacturing*, 12(2):101–119, 2004.

[37] W. J. Forrester. Industrial dynamics—a major breakthrough for decision makers. *Harvard Business Review,*, 36(4):37–66, 1958. Cambridge, Mass.

[38] M.S. Fox, M. Barbuceanu, and R. Teigen. Agent-oriented supply-chain management. *International Journal of Flexible Manufacturing Systems*, 12:165–188(24), 2000.

[39] Yonghui Fu, Rajesh Piplani, Robert de Souza, and Jingru Wu. Agent-based simulation applications: multi-agent enabled modeling and simulation towards collaborative inventory management in supply chains. In *WSC '00: Proceedings of the 32nd conference on Winter simulation*, pages 1763–1771, San Diego, CA, USA, 2000. Society for Computer Simulation International.

[40] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1983. Chapter 1, Section 2.4.

[41] M. Georgeff. A theory of action for multiagent planning. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 205–209. Kaufmann, San Mateo, CA, 1988.

[42] Khaled Ghoneim and Jamal Bentahar. An adaptive configuration for agent-based supply network. In *ITNG '08: Proceedings of the Fifth International Conference on Information Technology: New Generations*, pages 1268–1271, Washington, DC, USA, 2008. IEEE Computer Society.

[43] A. Gmez. Knowledge sharing and reuse. In J. Liebowitz, editor, *The Handbook of Applied Expert Systems*. CRC Press, 1998.

[44] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Aquisition*, 5:199–220, 1993.

[45] Peter Fingar Howard Smith. Business process management: The third wave. Technical report, The Business Process Management Intiative and the Workflow Management Coalition, 2003.

[46] M.N. Huhns and M.P. Singh. *Agents and Multi-agent Systems: Themes, Approaches, and Challenges.*, pages 1–23. Morgan Kaufmann, San Francisco, Calif., 1998.

[47] T. Ishida, L. Gasser, and M. Yokoo. Organization self-design of distributed production systems. *IEEE Trans. on Knowl. and Data Eng.*, 4(2):123–134, 1992.

[48] Kenn Steger-Jensen Jacques Trienekens, Hans-Henrik Hvolby and Peter Falster. Architectural frameworks for business information system analysis and design. In *Lean Business Systems and Beyond*, volume 257/2008 of *IFIP International Federation for Information Processing*, pages 413–421. Springer Boston, 2008.

[49] Nicholas R. Jennings. Coordination techniques for distributed artificial intelligence. In Nicholas R. Jennings and G. M. P. O'Hare, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley, 1996.

[50] Nicholas R. Jennings. Agent based computing: Promise and perils. In *Proceedings 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), Stockholm , Sweden*, pages 1429–1436, 1999.

[51] Nick Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[52] Nick Jennings and Michael Wooldridge. Applications of intelligent agents. In *Agent*

technology: Foundations, applications and markets, pages 3–28. Springer-Verlag New York, Inc., 1998.

[53] Nick R. Jennings, Pablo Noriega, and Simon Parsons. A framework for argumentation-based negotiation. pages 177–192, 1998.

[54] Li Jiangand and Da-You Liu. Advances in multi-agent coordination. In *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, pages 125–130, Dalian, 2006.

[55] (Roger)Jiao Jianxin, Xiao You, and Kumar Arun. An agent-based framework for collaborative negotiation in the global manufacturing supply chain network. *Robotics and Computer-Integrated Manufacturing*, 22(3):239–255, 2006.

[56] Albert Jones, Nenad Ivezic, and Michael Gruninger. Toward self-integrating software applications for supply chain management. *Information Systems Frontiers*, 3(4):403–412, 2001.

[57] C. Kafoglis. Maximize competitive advantage with a supply chain vision. *Hydrocarbon Processing*, 78(2):4750, 1999.

[58] T. Kaihara. Multi-agent based supply chain modelling with dynamic environment. *International Journal of Production Economics*, 85(2):263–269, 2003.

[59] Omar Kallel, Ins Ben Jafar, Lionel Dupont, and Khaled Ghedira. Multi-agent negotiation in a supply chain - case of the wholsale price contract. In Jos Cordeiro and Joaquim Filipe, editors, *ICEIS (4)*, pages 305–314, 2008.

[60] Elizabeth A. Kendall. Role modeling for agent system analysis, design, and implementation. *IEEE Concurrency*, 8(2):34–41, 2000.

[61] B. Kigler, C. Reuter. *Collaborative planning*, pages 223–240. Springer, 2002.

[62] Paulo Leito and Francisco Restivo. A framework for distributed manufacturing applications. In *In proceedings of the 2000 Advanced Summer Institute (AIS*, pages 1–7, 2000.

[63] T. Levitt. Exploit the product life cycle. *Harvard Business Review*, November - December:81–94, 1965.

[64] R. Lewicki, D. Saunders, J. Minton, and B. Barry. *Essentials of Negotiation*. McGraw Hill Higher Education, 2003.

[65] Bang-Qing Li, Jian-Chao Zeng, Meng Wang, and Gui-Mei Xia. A negotiation model through multi-item auction in multi-agent system. *Machine Learning and Cybernetics, 2003 International Conference on*, 3:1866–1870, Nov. 2003.

[66] Ye Li, Yan Chen, and Jia-Huan Li. Research on dynamic supply chain network. In *ICEC '05: Proceedings of the 7th international conference on Electronic commerce*, pages 379–384, New York, NY, USA, 2005. ACM.

[67] K.C. Lin and P. Lu. Inverse simulation – an error analysis. *Simulation*, 65(6):385–392, 1995. Fra Thz.

[68] Fernando Lopes, Nuno Mamede, A.Q. Novais, and Helder Coelho. Negotiation in a multi-agent supply chain system. 2002.

[69] Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Comput. Surv.*, 26(1):87–119, March 1994.

[70] Frank-Walter Jaekel Markus Rabe, Frank-Walter Jaekel. Reference models for supply chain design and configuration. In J. Liu B. G. Lawson D. M. Nicol L. F. Perrone, F. P. Wieland and R. M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*, 2006.

[71] C. E. Martin, K. S. Barber, and K. S. Barber. Agent autonomy: Specification, measurement, and dynamic adjustment. In *In Proceedings of the Autonomy Control Software Workshop, Agents '99*, pages 8–15, 1999.

[72] Francisco Maturana, Weiming Shen, and Douglas Norrie. Metamorph: an adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37(10):2159–2173, 1999.

[73] Nicolas Maudet. Negotiating dialogue games. *Autonomous Agents and Multi-Agent Systems*, 7(3):229–233, 2003.

[74] R.P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25(2):699738, 1987.

[75] M.J. Meixell and V.B. Gargeya. Global supply chain design: a literature review and critique. *Transportation Research Part E: Logistics and Transportation Review*, 41:531–550, 2005.

[76] Martin Steele Olivier de Weck Michael Galluzzi, Edgar Zapata. Foundations of supply chain management for space application. In *Proc. of 44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.

[77] I. Minakov, G. Rzevski, P. Skobelev, and S. Volman. Adaptive planning for supply chain networks. volume 4659 of *Lecture Notes in Computer Science*, pages 215–224, Regensburg, Germany, September 2007.

[78] Thierry Moyaux, Peter McBurney, and Michael Wooldridge. A supply chain as a network of auctions. Technical Report ULCS-07-012, Department of Computer Science, University of Liverpool, UK, July 2007.

[79] Roger B. Myerson. Mechanism design by an informed principal. *Econometrica*, 51(6):1767–1797, 1983.

[80] Serguei Netessine and Nils Rudi. Supply chain choice on the internet. *Manage. Sci.*, 52(6):844–864, 2006.

[81] Yew-Kwang Ng. *Welfare Economics*. Macmillan, 1983.

[82] Mark E. Nissen. Agent-based supply chain integration. *Information Technology and Management*, pages 289–312, 2001.

[83] Chris Nøkkentved. *Enabling Supply Networks with Collaborative Information Infrastructures*. PhD thesis, Copenhagen Business School, 2007.

[84] A. Norrman and U. Jansson. Ericsson's proactive supply chain risk management approach after a serious sub-supplier accident. *International Journal of Physical Distribution & Logistics Management*, 34(5):434–456, 2004.

[85] Juha-Miikka Nurmilaakso. Xml-based supply chain integration: A review and a case study. Master's thesis, HELSINKI UNIVERSITY OF TECHNOLOGY, 2003.

[86] H.S. NWANA and D.T. NDUMU. A perspective on software agents research. *The Knowledge Engineering Review*, 14(02):125–142, 1999.

[87] Hyacinth Nwana, Lee Lyndon, and Nick Jennings. Co-ordination in software agent systems. *The British Telecom Technical Journal*, 14(4):79–88, 1996.

[88] James H. Odell, H. van Dyke Parunak, and Bernhard Bauer. Extending uml for agents. In *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence*, pages 3–17, 2000.

[89] Osborne and Rubinstein. *A course in Game Theory*. MIT Press, 1994.

[90] Young pa So and Edmund H. Durfee. Designing tree-structured organizations for computational agents. *Computational and Mathematical Organization Theory*, 2:219–246, 1996.

[91] H. Van Dyke Parunak and James Odell. Representing social structures in uml. In *AOSE '01: Revised Papers and Invited Contributions from the Second International Workshop on Agent-Oriented Software Engineering II*, pages 1–16, London, UK, 2002. Springer-Verlag.

[92] S. Pathak, G. Nordstorm, and S. Kurokawa. Modelling of supply chain a multi-agent approach. Nshville,Tennesse, October 2000. IEEE SMC 2000.

[93] H. Edward Pattison, Daniel D. Corkill, and Victor R Lesser. Instantiating descriptions of organizational structures. In Michael N. Huhns, editor, *Distributed Artificial Intelligence, Research Notes in Artificial Intelligence*, volume I, pages 59–96. Pitman Publishers, 1987.

[94] Alexander Pokahr and Lars Braubach. Jadex user guide, 2007. (accessed on september,14,2008).

[95] Iyad Rahwan. Argumentation in multi-agent systems. *.Autonomous Agents and Multi-Agent Systems*, 11(2):115–125, 2005.

[96] Iyad Rahwan, Ryszard Kowalczyk, and Yun Yang. Virtual enterprise design - bdi agents vs. objects. *Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader*, 2112:147–157, 2001.

[97] Iyad Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.

[98] Iyad Rahwan, Liz Sonenberg, and Peter McBurney. Bargaining and argument-based negotiation. In *ArgMAS 2004*, pages 176–191, 2004.

[99] A. S. Rao and M. P. Georgeff. Bdi agents: from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems*, San Francisco, 1995.

[100] Avi Rosenfeld, Gal A. Kaminka, Sarit Kraus, and Onn Shehory. A study of mechanisms for improving robotic group performance. *Artif. Intell.*, 172(6-7):633–655, 2008.

[101] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge (Massachusetts, USA), 1994.

[102] Stuart Russell and Peter Norvig. *Artificial Intelligence - A modern approch*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[103] G. Rzevski, P. Skobelev, S. Batishchev, and A. Orlov. *A Framework for Multi-Agent Modelling of Virtual Organisations*, pages 253–260. Kluwer Academic Publishers, 2003.

[104] Norman Sadeh, David Hildum, Dag Kjenstad, and Allen Tseng. MASCOT: An agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In *Workshop notes, Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, 3rd International Conference on Autonomous Agents (Agents' 99)*, 1999.

[105] Tuomas Sandholm. Automated negotiation. *Commun. ACM*, 42(3):84–85, 1999.

[106] Tuomas Sandholm. Approaches to winner determination in combinatorial auctions. *Decis. Support Syst.*, 28(1-2):165–176, 2000.

[107] SAP. Adaptive supply chain networks. SAP White Paper, 2002. (accessed on Aug,20,2008).

[108] Supply Chain Council SCC. SCOR 8.0 overview booklet, 2007. Available from http://www.supply-chain.org/ (accessed March 2007).

[109] J.S. Shafritz, J.M. & Ott. *Classics of organization theory.* Wadsworth Publishing Company, 1996.

[110] Mildred L G Shaw and Brian R Gaines. Knowledge support systems for constructively channelling conflict in group dynamics. In *AAAI-94 Workshop on models of,* pages 107–116. AAAI Press, 1994.

[111] M. Shoham, Y. & Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceeedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, 1992.

[112] Jaime Sichman, Rosaria Conte, Yves Demazeau, and Cristiano Castelfranchi. A social reasoning mechanism based on dependence networks. In A. G. Cohn, editor, *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI '94)*, pages 188–192. John Wiley & Sons Ltd., 1994.

[113] David Simchi-Levi, Philip Kaminsky, and Edith Simchi-Levi. *Designing and Managing the Supply Chain.* McGraw-Hill Higher Education, 2000.

[114] Z. Skolicki and T. Arciszewski. Intelligent agents in design. In *Proceedings of 2003 ASME International Design Engineering Technical Conferences & The Computers and Information in Engineering Conference*, Chicago, 2003.

[115] A. Slade and A. Bokma. Ontologies within extended enterprises. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 1*, page 39.1, Washington, DC, USA, 2002. IEEE Computer Society.

[116] Adam Smith. *An inquiry into the nature and causes of the wealth of nations.* Clarendon Press, Oxford, UK, 1976.

[117] Adam Smith. *The Theory of Moral Sentiments (Great Books in Philosophy)*. Prometheus Books, May 2000.

[118] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *Transactions on Computers*, C-29(12):1104–1113, 1980.

[119] J. Swaminathan, N. Sadeh, and S. Smith. A knowledge-based multi-agent simulation testbed to support supply chain design and management decisions. In *Proceedings IJCAI-93 Workshop on Knowledge-Based Production Planning, Scheduling and Control*, 1993.

[120] T. Tetenbaum. Shifting paradigms: from newton to chaos. *Organisational Dynamics*, 26(4):21–32, 1998.

[121] A. Del Towill, D. R.and Vecchio. The application of filter theory to the study of supply chain dynamics. *International Journal of Production Planning and Control*, 5(1):82–96, 1994.

[122] D. R. Towill. Industrial dynamics modelling of supply chains. *International Journal of Physical Distribution & Logistics Management*, 26(2):23–42, 1996.

[123] D. R. Towill. The seamless supply chain - the predator's strategic advantage. *International Journal of Technology Management*, 13(1):37–56, 1997.

[124] Tu Hoang Truong and Farhad Azadivar. Simulation optimization in manufacturing analysis: simulation based optimization for supply chain configuration design. In *WSC '03: Proceedings of the 35th conference on Winter simulation*, pages 1268–1275. Winter Simulation Conference, 2003.

[125] Andy A. Tsay, Steven Nahmias, and Narendra Agrawal. Modelling supply chain contracts: A review. In Tayur et al., editor, *Quantitative Models for Supply Chain Management*, pages 299–336. Kluwer Academic Publishers, Dordrecht, 1999.

[126] Stephen J. Turner, Wengton Cai, Wen-Jing Hsu, Boon-Ping Gan, Li Liu, and Sanjay Jain. Distributed supply chain simulation across enterprise boundaries. *Proceedings of the 2000 Winter Simulation Conference*, 2000.

[127] F.H. van Eemeren and R. Grootendorst. *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Lawrence Erlbaum Assoc Inc, 1996.

[128] Hal R. Varian. Economic mechanism design for computerized agents. In editor Usenix, editor, *Proc. Usenix Conference on Electronic Commerce (New York, NY, USA)*, 1995.

[129] VICS Association. Cpfr, 2008. (accessed 10 September 2008).

[130] Christoph Haehling von Lanzenauer and Karsten Pilz-Glombik. Coordinating supply chain decisions: an optimization model. *OR Spectrum*, 24:59–78, 2002.

[131] Mark A. Vonderembse, Mohit Uppal, Samuel H. Huang, and John P. Dismukes. Designing supply chains: Towards theory development. *International Journal of Production Economics*, 100(2):223–238, April 2006.

[132] E. C. W. Walton, D. N.& Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany, New York, 1995.

[133] Yan-Ling Wang and De-Li Yang. A heuristic method for logistics supply chain coordination and risk control. *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on*, 2:866–870, July 2006.

[134] Jack F. Williams. A hybrid algorithm for simultaneous scheduling of production and distribution in multi-echelon structures. *Management Science*, 29(1), 1983.

[135] Michael Wooldridge. *An Introduction to Multi-Agent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[136] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[137] Heekwon Chae Younghwan Choi, Dongwoo Kang and Kwangsoo Kim2. An enterprise architecture framework for collaboration of virtual enterprise chains. *The International Journal of Advanced Manufacturing Technology*, 35:1065–1078, 2008.

[138] Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Organizational abstractions for the analysis and design of multi-agent system. In *First international workshop, AOSE 2000 on Agent-oriented software engineering*, pages 235–251, Secaucus, NJ, USA, 2001. Springer-Verlag New York, Inc.

Appendix A

Simulation replications outputs without negotiation

Table 7: Simulation replications without negotiation

a	b	c	d	e	f	g
1	B	87	Dsitributor0	73	Supplier0	1.28
2	F	0	0	0	0	0
3	B	80	Distributor1	67	Supplier5	1.85
4	F	0	0	0	0	0
5	F	0	0	0	0	0
6	F	0	0	0	0	0
7	B	86	Distributor0	72	supplier2	1.25
8	F	0	0	0	0	0
9	F	0	0	0	0	0
10	F	0	0	0	0	0
11	F	0	0	0	0	0
12	F	0	0	0	0	0
13	F	0	0	0	0	0
14	B	84	Distributor1	70	supplier3	1.84

Continued on Next Page...

Table 7 – Continued

a	b	c	d	e	f	g
15	B	76	Distributor2	64	supplier6	0.65
16	B	91	Distributor0	76	supplier2	1.85
17	F	0	0	0	0	0
18	B	108	Distributor2	80	Supplier8	1.88
19	B	96	Distributor0	80	Supplier2	1.85
20	B	86	Distributor0	72	Supplier0	0.98
21	B	88	Distributor2	74	Supplier7	0.96
22	B	88	Distributor1	74	Supplier3	1.25
23	B	108	Distributor1	90	Supplier3	1.56
24	B	81	Distributor0	68	Supplier1	0.65
25	B	92	Distributor1	77	Supplier4	1.55
26	B	84	Distributor1	70	Supplier5	1.28
27	B	86	Distributor0	72	Supplier1	1.58
28	F	0	0	0	0	0
29	B	76	Distributor0	64	Supplier0	1.55
30	B	0	0	0	0	0

[a]Number of replications.
[b]Supply network formation results "Bid/Fail".
[c]The distributer's winning Price.
[d]The winning distributor.
[e]The supplier's winning price.
[f]The winning supplier.
[g]The customer's gained utilities Value.

Appendix B

Simulation Replications outputs with Negotiation

Table 8: Simulation replications with negotiation

a	b	c	d	e	f	g
1	B	87	Dsitributor0	73	Supplier0	1.28
2	N	105	Distributor3	91	Supplier6	1.54
3	B	80	Distributor1	67	Supplier5	1.85
4	N	92	Distributor0	80	Supplier0	1.84
5	N	106	Distributor0	91	Supplier0	1.84
6	F	0	0	0	0	0
7	B	86	Distributor0	72	supplier2	1.25
8	N	88	Distributor2	77	Supplier4	1.84
9	N	119	Distributor0	101	Supplier0	1.85
10	N	97	Distributor2	84	Supplier8	1.56
11	N	105	Distributor1	91	Supplier5	1.86
12	F	0	0	0	0	0
13	N	121	Distributor1	104	Supllier3	1.85
14	B	84	Distributor1	70	supplier3	1.84

Continued on Next Page...

Table 8 – Continued

a	b	c	d	e	f	g
15	B	76	Distributor2	64	supplier6	0.65
16	B	91	Distributor0	76	supplier2	1.85
17	N	96	Distributor2	84	Supplier6	1.88
18	B	108	Distributor2	80	Supplier8	1.88
19	B	96	Distributor0	80	Supplier2	1.85
20	B	86	Distributor0	72	Supplier0	0.98
21	B	88	Distributor2	74	Supplier7	0.96
22	B	88	Distributor1	74	Supplier3	1.25
23	B	108	Distributor1	90	Supplier3	1.56
24	B	81	Distributor0	68	Supplier1	0.65
25	B	92	Distributor1	77	Supplier4	1.55
26	B	84	Distributor1	70	Supplier5	1.28
27	B	86	Distributor0	72	Supplier1	1.58
28	N	88	Distributor1	77	Supplier5	1.86
29	B	76	Distributor0	64	Supplier0	1.55
30	F	0	0	0	0	0

[1]Number of replications.
[2]Supply network formation results "Biding/Negotiation/Fail".
[3]The distributer's winning Price.
[4]The winning distributor.
[5]The supplier's winning price.
[6]The winning supplier.
[7]The customer's gained utilities Value.

Index