Integrative Negotiation In Complex Organizational Agent Systems[*]

Xiaoqin Zhang
Computer and Information Science Department
University of Massachusetts at Dartmouth
x2zhang@umassd.edu

Victor Lesser
Computer Science Department
University of Massachusetts at Amherst
lesser@cs.umass.edu

Tom Wagner
Automated Reasoning Group
Honeywell Laboratories
Tom.Wagner@honeywell.com

ABSTRACT

This paper addresses the problem of negotiation in a complex organizational context and tries to bridge the gap between self-interested negotiation and cooperative negotiation. An integrative negotiation mechanism is introduced, which enables agents to choose any attitude from the extremes of self-interested and cooperative to those that are partially self-interested and partially cooperative. This mechanism is based on and also extends the motivational qualities(MQ) framework for evaluating which task an agent should pursue at each time point. Experimental work verifies this mechanism and explores the question whether it always improves the social welfare to have an agent be completely cooperative.

Keywords

integrative negotiation, motivation, goal selection & theories, group and organizational dynamics

1. INTRODUCTION

In Multi-Agent systems (MAS), agents negotiate over task allocation, resource allocation and conflict resolution problems. Until now almost all related work on negotiation can be categorized into two general classes: cooperative negotiation and competitive negotiation. In competitive negotiation, agents are self-interested and negotiate to maximize their own local utility; in cooperative negotiation, agents work to find a solution that increases their joint utility – the sum of the utilities of all involved agents. In the competitive negotiation class, significant work [6] has been done in the area of bounded rational self-interested agents (BRSI). Said agents are self-interested and social welfare is not a concern – each agent works to maximize its own utility though contracting, bidding and decommiting. In the cooperative negotiation class, significant work has been done in the area of conflict resolution through negotiation [5]. In these approaches there is no notion of individual agent utility – agents are "completely-cooperative" with each other and cooperate to solve problems together. Little work has been done to study negotiation between these two extreme cases.

We feel that as the sophistication of multi-agent systems increases, MAS will be neither simple market systems where each agent is purely self-interested, seeking to maximize its local utility, nor distributed problem solving systems where all agents are completely cooperative working to maximize their joint utility. This will occur for two reasons. First, agents from different separate organizational entities will come together to dynamically form virtual organizations/teams for solving specific problems that are relevant to each of their organizational entities [3]. How these agents work in their teams will often be dependent on the existence of both long term and short-term relationships and conform to their underlying organizational entities. Also, even for agents from self-interested organizations, it might be beneficial for them to be partially cooperative when they are in the situations where they will have repeated transactions with other agents from other organizational entities. Additionally, agents may be involved concurrently with more than one virtual organization while doing tasks for their own organizational entities.

Secondly, even agents working solely with agents of their own organizational entities will take varying attitudes in the spectrum of fully cooperative to totally self-interested in order for the organization to best achieve its overall goal. This perspective is based on a bounded-rational argument: it is not possible from a computational or communicational perspective for an agent to be fully cooperative, because the agent needs to take into account the utilities of all agents in the organization and the state of achievement of all organizational goals to be fully cooperative. Thus, it may be best for the organization to have agents being partially cooperative in their local negotiation with other agents rather than being fully cooperative in order to deal more effectively with the uncertainty of not having a more informed view of the state of the entire agent organization.

Given the complex organizational context in multi-agent systems, it is not enough for an agent simply to be purely self-interested or completely cooperative. It needs to have more flexible negotiation strategies between these two extreme cases, i.e. half-self-

[*] This material is based upon work supported by the National Science Foundation under Grant No.IIS-9812755 and the Air Force Research Laboratory/IFTD and the Defense Advanced Research Projects Agency under Contract F30602-99-2-0525. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory/IFTD, National Science Foundation, or the U.S. Government.

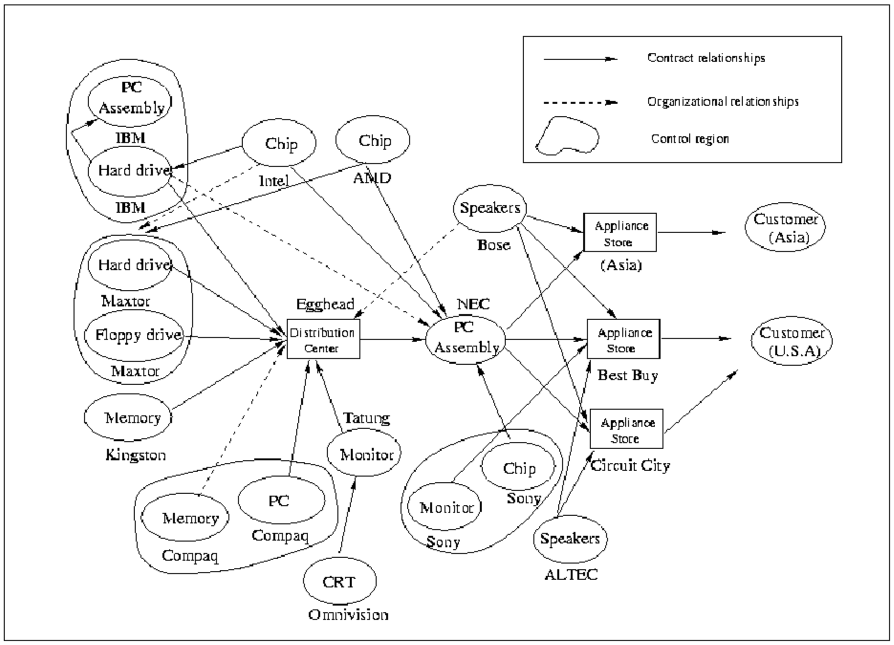

Figure 1: Supply chain example

interested, half-cooperative; mostly self-interested, slightly cooperative; etc. For example, let's consider the supply chain example in Figure 1. There are different organizational relationships among agents. For instance, there is an agent (agent_IBM_2) who produces hard drives, belonging to the IBM company. It provides hard drives for three different agents, with the following organizational relationships to it:

1. Agent_IBM_2 provides hard drives for the other agent (agent_ IBM_ 1) that also belongs to IBM but assembles PC.
2. Agent_IBM_2 provides hard drives to an NEC agent (agent_ NEC), and as the transactions between them become more frequent and regular, they form a virtual organization based on the recent transactions.
3. Agent_IBM_2 occasionally also provides hard drives for a distributor center (agent_DIS) based on a simple marketing mechanism.

When agent_IBM_2 negotiates with these three agents, it should use different negotiation strategies. When it negotiates with agent_IBM_1, it may need to be more cooperative than it is towards the other two agents if its most important goal is to increase the utility of IBM. However, even for the good of IBM's benefit, it may not be the best choice for agent_IBM_2 always to be accommodative towards agent_IBM_1. Sometimes it may bring IBM more profit for agent_IBM_1 to provide hard drives to agent_DIS rather than to agent_IBM_1. So the question is: how cooperative should agent_IBM_2 be towards agent_IBM_1?

When agent_IBM_2 negotiates with agent_NEC_1, it may need to be more cooperative than it is towards agent_DIS given the virtual organization it has formed with agent_NEC_1. How cooperative it should be depends on how important the utility increase of this virtual organization is to agent_IBM_2 and how the goal to increase the utility of this virtual organization relates to its other goals. Also, as we noticed before, the formation of this virtual organization is dynamic; it may also disappear sometime later if the environment changes, so agent_IBM_2 should adopt its negotiation strategy dynamically too.

When agent_IBM_2 negotiates with agent_DIS, should it be purely self-interested, given that there is only a simple marketing relationship between them? It may not be the case. There are two reasons. First, when there are multiple attributes involved in negotiation, it is possible to reach a win-win agreement rather than play a zero-sum game. Secondly, it is not good for an agent to try to maximize its own utility in every negotiation session from a long-term perspective; in other words, it may lose its long-term profit by being too aggressive every time (do you want to go back to the same sales person who tries to squeeze every penny out of your pocket?).

From the above examples we find it is necessary to have the following framework to support an agent's negotiation in a complex organizational context:

1. The agent can choose from many different negotiation strategies in the spectrum from purely self-interested to accommodative. It should be easy for the agent to switch from one strategy to another strategy. A uniformed negotiation mechanism for all different negotiation strategies is an ideal choice.
2. The choice of negotiation strategy should not be hard-coded in the agent. The choice should depend on the agent's organizational goals, the current environmental circumstance, which agent is negotiated with, and what issue is under negotiation.
3. There should be no requirement of a centralized controller which coordinates the agent's behavior. The agent should be free to choose any negotiation strategy according to its goals set by its designer. Because every agent belongs to different users, it is not realistic to assume a centralized controller.

So far, there has been no such negotiation framework which provides the above capabilities for agents (see related work in Section 6). In this paper, we introduce an integrative negotiation mechanism which enables agents to manage all sorts of negotiation strategies in the spectrum from self-interested to completely cooperative in a uniform reasoning framework called the MQ framework. This mechanism is based on the Motivational Quantities (MQ) framework [8], which is reviewed in Section 2. Section 3 describes the integrative negotiation mechanism. Section 4 uses examples to explain the ideas. Section 5 presents experimental results that explore how different negotiation attitudes affect the agent's performance and the social welfare of the overall system. Section 6 discusses related work and Section 7 concludes and identifies further work.

2. MQ FRAMEWORKS

The MQ framework is an agent control framework that provides the agent with the ability to reason about which tasks should be performed and when to perform them. The reasoning is based on the agent's organizational concerns. The basic assumption is that agents are complex, with multiple goals related to the multiple roles they play in the agent society. The progress toward one goal can not substitute for the progress toward another goal. Motivational (MQs) are used to represent the progress toward organizational goals quantitatively.

Each agent has a set of MQs which it is interested in and wants to accumulate. Each MQ_i in this set represents the progress toward one of the agent's organizational goals. Each MQ_i is associated with a preference function (utility curve), U_{f_i}, that describes the agent's preference for a particular quantity of the MQ_i. Different types of MQs are not interchangeable. The utility U_i associated with MQ_i can not be transferred to the utility U_j associated with MQ_j if $i \neq j$. The agent's overall utility is a function of the different utilities associated with the MQs it tracks: $U_{agent} = \gamma(U_i,\ U_j,\ U_k,\ ..)$. The structure of function γ represents the agent's preference and emphasis on different organizational goals. The MQ framework thus provides an approach to compare the agent's different motivational factors through a multi-attribute function.

MQs are consumed and produced by performing MQ tasks. The agent's overall goal is to select tasks to perform in order to maximize its local utility through collecting different MQs. This does not mean that the agent has to be "self-interested"; it only means that the agent selects its actions to contribute to its multiple organization goals. If "to help agent B" is one of the goals of agent A, then agent A will act as a cooperative to agent B. If two or more agents have a goal in common and hence have the same MQ in common, they act as a group or a team working collaborately toward this goal.

MQ tasks are an abstraction of the primitive actions that an agent may perform. The agent compares and selects tasks that are associated with different organizational goals. Each MQ task T_i has the following characteristics:

- Earliest start time (est), $start_i$. The performance of T_i before this time does not generate valid results.
- Deadline, $deadline_i$. The accomplishment of T_i after this time does not generate valid results.
- MQ task T_i needs some process time to be accomplished, denoted as d_i.
- MQ task T_i produces certain quantities of one or more MQs, denoted as $MQPS$ (MQ production set). The production of MQs reflects the progress made by the accomplish of the task toward a goal.
- MQ task T_i consumes certain quantities of one or more types of MQs, denoted as $MQCS$ (MQ consumption set). The consumption of MQs represents resources consumed by performing this task, or favors owed to other agents for subcontracting work.

The MQ scheduler schedules current potential MQ tasks, and produces a schedule of a set of MQ tasks, specifying their start times, finish times. The scheduler takes the following factors into consideration: the $MQPS$, $MQCS$, duration d_i, the earliest start time and the deadline of each MQ task.

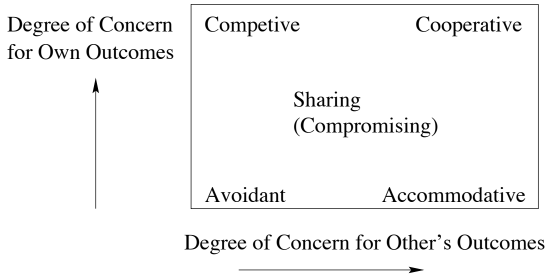

Figure 2: The dual concern model

3. INTEGRATIVE NEGOTIATION

In a complex agent society, an agent will need to work with other agents from a variety of different organizational positions. For example, an agent from its own group, an agent who has a higher position and thus more authority, an agent from a cooperative company, or an agent from a competing company and so forth. The agent's attitude toward negotiation is not just simply either competing or cooperative, the agent needs to qualitatively reason about each negotiation session, e.g., how important its own outcome is compared to the other agents' outcomes, so it can choose an appropriate negotiation strategy.

Figure 2 describes this dual concern model [2]. When the agent only attaches importance to its own outcome, its attitude toward negotiation is competitive (self-interested); when an agent attaches the same degree of importance to its own outcome as it does to the outcomes of the other agent, its attitude is cooperative; when the agent attaches more importance to the outcomes of other agents and no importance to its own outcome, its attitude is accommodative; if the agent attaches no importance to any outcomes, its attitude is avoidant (the negotiation is not worth its time and effort). From this model, we find that there are potentially many options between the two extremes of self-interested and cooperative. These other options depend on the importance the agent attaches to the increase of its own utility relative to the importance it attaches to the other agents' utility increases.

The MQ framework can support sophisticated negotiation where each negotiation issue has MQ transference associated with it. Let's use task allocation as an example of negotiation where for each task *t* allocated to agent B, from agent A, certain MQs are transferred from agent A to agent B. The conceptual model here is that agent B is motivated by the potential increase in its MQs to perform tasks for agent A (note that this does not convert the MQs to currency as not all agents may be interested in said MQs). We will start with a simple, abstract example. In this model, when agent B commits to accomplishing task *t*, based on a contract that is mutually agreed upon by the two agents (formed either dynamically or pre-defined), it is then obligated to perform the task. When B successfully accomplishes *t*, the agreed upon amount of the MQ will be transferred from agent A to agent B. Note that agent B must actually decide whether or not it is interested in performing *t*. This evaluation is done via the MQ framework and the associated MQ scheduler. The evaluation uses agent B's preference for the MQ in question to determine the the relative value of performing *t* for agent A. This valuation process, in turn, determines agent B's attitude toward the negotiation of task *t*.

In terms of specifics, there are two types of MQs that could be transferred with the successful accomplishment of task *t*: *goal_related*

MQ and *relational MQ*. These classes are conceptual and used to clearly differentiate motivations for task performance from attitudes toward negotiation issues – in reality, they are both simply MQs. *Goal_related MQ*s are associated with an agent's organizational goals and generally increases in MQ volume have positive benefits to the agent's utility. Note that the agent's designer determines which kinds of MQs the agent tracks (and is interested in), defines the agent's preference for each via the utility functions discussed earlier, and determines how these relate to the agent's organizational goals. When dealing with *goal_related MQ*s, the agent collects MQs for its own utility increase. In this sense, agent B's performance of task *t* is motivated by "self-interested" reasons if payment is via a *goal_related MQ*. For example, task *t* has 3 units of MQ_x transferred with it, and for agent B, the utility curve of MQ_x is: u(x) = 2x, that means, the utility of agent B will increase by 6 units by collecting 3 units of MQ_x though performing task *t*. Agent B decides whether to accept task *t* by reasoning about its value relative to the cost of the resources it will expend in the performance of *t*. In this case, as the task doesn't consume any MQs, the resource expenditure is time or in terms of opportunity cost. Because this reasoning process pertains to *goal_related MQ*s, it is "self-interested" for the agent's only concerns is its own utility increase.

Consider a modified case. Suppose that by having task *t* accomplished agent A's own utility increases by 20 units. If agent B takes this fact into consideration when it makes its decision about task *t*, agent B is cooperative with agent A because agent B is also concerned about agent A's outcome (in addition to its own). If we want agent B to consider A's utility, we need to introduce another MQ designed to model B's (revised) preference for A to have a utility increase also. To reflect the B's attitude toward A's outcome, we introduce a *relational MQ*, the preference for which represents how cooperative agent B is with agent A concerning task *t*. Let $MQ_{ba/t}$ be the relational MQ transferred from agent A to agent B when agent B performs task *t* for agent A. Since $MQ_{ba/t}$ is a *relational MQ*, its only purpose is to measure the relationship between agents A and B. When measuring the utility of agent B toward problem solving, we will not consider the utility produced by any relational MQs such as $MQ_{ba/t}$. Likewise with agent A. When agent A transfers $MQ_{ba/t}$ to agent B, we will not tabulate the negative change in utility of agent A because the change in utility is not related to problem solving progress but is instead related to the transfer of a relational MQ. The reason for this approach is that in this paper our performance metric is social welfare as it is conventionally used, which is in terms of progress toward joint goals. From this view, the utility produced by a relational MQ can be seen as *virtual utility*. Though $MQ_{ba/t}$ produces virtual utility, is important because it carries the information of how important task *t* is for agent A[1] and makes it possible for agent B to consider agent A's outcome when it makes its own decisions. Actually, how $MQ_{ba/t}$ is mapped into agent B's (virtual) utility, meaning utility that is not included in the social welfare computation[2] depends on how cooperative agent B is with agent A. Suppose that 20 units $MQ_{ba/t}$ are transferred with task t, representing the utility agent A gained by having agent B perform task t, Figure 3 shows four different functions for mapping $MQ_{ba/t}$ to agent B's utility.

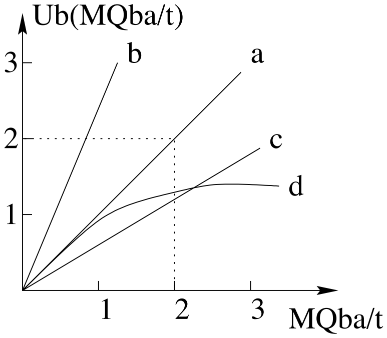

Figure 3: different mapping functions of $MQ_{ba/t}$

Function a, b and c are linear functions: $U_a(MQ_{ba/t}) = k * MQ_{ba/t}$.

If $k = 1$(a), $U_b(MQ_{ba/t}) = MQ_{ba/t} = U_a(t)$ ($U_a(t)$ denotes the utility agent A gained by transferring t), then agent B is completely cooperative to agent A[3];

If $k > 1$(b), $U_b(MQ_{ba/t}) > MQ_{ba/t} = U_a(t)$, then agent B is accommodative to agent A;

If $k < 1$(c), $U_b(MQ_{ba/t}) < MQ_{ba/t} = U_a(t)$, then agent B is partially cooperative with agent A;

If $k = 0$, $U_b(MQ_{ba/t}) = 0$, then agent B is self-interested with respect to agent A. In this case, if agent A wants agent B to do task t, it needs to transfer another kind of MQ (the *goal_related MQ*) to agent B, agent B and agent A can negotiate about what type of *goal_related MQ* to transfer and how much of it should be transferred, regarding how and when agent B could accomplish task t. In the following examples and the experimental work, we assume that the type and amount of the transferred *goal_related MQ*s are fixed and agents do not negotiate about them, so we can demonstrate how the *relational MQ* works.

The mapping function could also be a nonlinear function (d) that describes a more complicated attitude of agent B to agent A, i.e., agent B being fully cooperative with agent A for some period and then becoming self-interested. An agent can adjust the utility mapping function to reflect its relationship with another agent, which could be it's administrator, colleague, friend, client or competitor. By adjusting some parameters in the mapping function, more subtle relationships could be managed. The agent could differentiate a friendly colleague from an unfriendly colleague, also it could draw distinctions between a best friend and an ordinary friend.

Different from the *goal_related MQ*s, which are built by the agent's designer and whose utility curves are not changing, the utility curves of the *relational MQ*s can be adjusted by the agent dynamically to reflect its dynamic relationships with other agents. The agent's attitude towards another agent could be "issue-specific"; given an agent could play multiple roles, there could be different issues negotiated between agents, and the agents should select different attitude according to what issue is negotiated. For example, for the colleague's request to contribute to a shared professional job and for the same colleague's request for a ride, even both requests come from the same agent, the agent's attitude could be different.

How can an agent choose its attitude toward other agents in such a complex organization context? We are not planning to present a solution to this question in this paper, but we feel that the agent should dynamically adjust its attitude by analyzing the other party, the issue in negotiation and its current problem-solving status. The

[1] It is assumed that agents are honest and don't lie about the importance of task *t*.

[2] In remainder of the paper, we may omit the word "virtual" before utility, but we know that this *relational MQ* only maps into virtual utility that is not real utility. In the experimental work, neither the agent's utility nor the social welfare includes the virtual utility from *relational MQ*

[3] It should be noticed that the relationship between agents is not symmetric, the fact that agent B is completely-cooperative to agent A does not imply that agent A is also completely-cooperative to agent B.

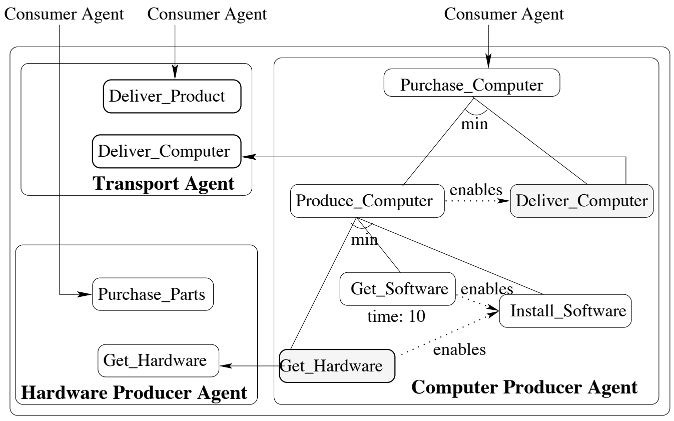

Figure 4: Agent Society

following information should be considered in this decision making process: "Who is the other agent?", "How is its organizational goals related to mine?", "What is its objective?", "What is its relationship to me?" and so forth. Some of this information can be learned from experience [7].

In the MQ framework, the MQ scheduler enables the agent to optimize its schedule and maximize it local utility. While the framework directly supports the concept of relational MQs and being motivated to cooperate on that basis, the use of MQ transference in this paper extends the MQ framework to interconnect the local scheduling problems of two or more agents in a dynamic fashion (based on the current context). Prior to this work, no meaningful work had been done in MQ transference or the implications of it.

4. THE SCENARIO

In this section, we introduce an example of a three-agent society and show how the integrative negotiation mechanism works using the MQ framework.

There are three agents in this society as shown in Figure 4:

1. Computer Producer Agent (*c*): receives "Produce_Computer" task from an outside agent (which is not considered in this example). Figure 4 shows that to accomplish "Produce_Computer" task, Computer Producer Agent needs to generate an external request for hardware ("Get_Hardware" task), and also needs to ship the computer ("Deliver_Computer") through a transport agent.
2. Hardware Producer Agent (*h*): receives task "Get_Hardware" from Computer Producer Agent, it also receives "Purchase_Parts" task from an outside agent.
3. Transport Agent (*t*): receives task "Deliver_Computer" from Computer Producer Agent, it also receives "Deliver_Product" task from an outside agent.

In this example, every agent collects the same type of *goal_related MQ*: "MQ_$". The utility curve for "MQ_$" is: $utility(x) = x$, every agent uses this same function. Each task that the agent receives includes following information:

- deadline (dl): the latest finish time for the task.
- reward (r): if the task is finished by the deadline, the agent will get reward r (which is r units of "MQ_$").
- early finish reward rate (e): If the agent can finish the task by the time (ft) as it promised in the contract, it will get the extra early finish reward: max (e*r*(dl-ft), r)[4] in addition to the reward r.

Hardware Producer Agent receives "Purchase_Parts" task from an outside agent with x units of MQ_$, where x is a random number varying from 2 to 10. Computer Producer Agent has long-term contract relationship with Hardware Producer Agent and Transport Agent: its "Get_Hardware" task always goes to Hardware Producer Agent with a fixed reward of 3 units of MQ_$, and its "Deliver_Product" task always goes to Transport Agent with a fixed reward of 3 units of MQ_$. Every "Produce_Computer" task comes to Computer Producer Agent with a reward of 20 units of MQ_$, if it is finished by its deadline, Computer Producer Agent would have its local utility increased by 14 units. Assume task "Get_Hardware" and "Deliver_Product" have the same importance, the accomplishment of each task would result in 7 units utility increase for Computer Producer Agent. This information is reflect by the 7 units of $MQ_{hc/t}$ transferred with task "Get_Hardware" and 7 units of $MQ_{tc/t}$ transferred with task "Deliver_Product". $MQ_{hc/t}$[5] is a *relational MQ* introduced to reflect the relationship of Hardware Producer Agent with Computer Producer Agent concerning task *t*. The transferred "MQ_*hc/t*" with the task represents the utility increase of Computer Producer Agent by having this task accomplished. How it is mapped into Hardware Producer Agent's virtual utility depends on Hardware Producer Agent's attitude towards the utility increase of Computer Producer Agent regarding task "Get_Hardware". If the "Produce_Computer" task could be finished earlier than its deadline, Computer Producer Agent could get more than 20 units reward. The extra utility increase could be estimated and reflected by more than 7 units transferred "MQ_*hc/t*" or "MQ_*tc/t*" for the other two agents. Suppose the following task is received by Computer Producer Agent:

task name : Purchase_Computer_A
est: 10
deadline: 70
reward: 20 units MQ_$
early finish reward rate: e=0.01

Through the reasoning of the MQ scheduler, Computer Producer Agent decides to accept it and finish it by time 40 (it leaves 4 units slack time) to earn extra early reward 6 $((70 - 40) * 0.01 * 20)$ units MQ_$. Its local utility increases by 20 units after the accomplishment of this task. Hence the following two task requests: Get_Hardware_A and Deliver_Computer_A are sent to Hardware Producer Agent and Transport Agent respectively:

task name	*Get_Hardware_A*	*Deliver_Computer_A*
est	*10*	*30*
deadline	*20*	*40*
reward	*3 units MQ_$* *10 units MQ_hc/t*	*3 units MQ_$* *10 units MQ_tc/t*
early finish reward rate[6]	*e=0.01*	*e=0.01*

In this example, we look at three different attitudes with a linear function: $U_{ha}(MQ_{hc/t}) = k * MQ_{hc/t}$.

1. k=1, Hardware Producer Agent is completely-cooperative with Computer Producer Agent regarding task "Get_Hardware".
2. k=0.5, Hardware Producer Agent is half-cooperative (partial cooperative) to Computer Producer Agent regarding task "Get_Hardware".
3. k=0, Hardware Producer Agent is self-interested to Computer Producer Agent regarding task "Get_Hardware".

Now we can look at how these different attitudes affect the negotiation process of Hardware Producer Agent. Suppose there are two other tasks "Purchase_Parts_A" and "Purchase_Parts_B" received

[4] For each time unit the task finishes earlier than the deadline, the contractee agent get extra reward is e*r, but the total extra reward would exceed the reward r.

[5] Similarly, $MQ_{tc/t}$ is a *relational MQ* that reflects the relationship of Transport Agent with Computer Producer Agent concerning task t. Detailed discussion about it is omitted here.

by Hardware Producer Agent besides task "Get_Hardware_A", following three tasks are sent to the MQ Scheduler (suppose the initial MQ set is empty):

task name	*Get_Hardware_A*	*Purchase_Parts_A*	*Purchase_Parts_B*
est	*10*	*10*	*10*
deadline	*20*	*30*	*20*
process time	*10*	*10*	*10*
MQPS	*[MQ_$,3]* *[MQ_hc/t, 10]*	*[MQ_$,4]*	*[MQ_$,9]*

If Hardware Producer Agent is completely-cooperative to Computer Producer Agent, the best MQ schedule produced is as following:
$[10, 20]$ `Get_Hardware_A` $[20, 30]$ `Purchase_Parts_A`
Hardware Producer Agent will have 7 units utility increase after the accomplishment of this schedule. If Hardware Producer Agent is self-interested to Computer Producer Agent, the best MQ schedule produced is as following:
$[10, 20]$ `Purchase_Parts_B` $[20, 30]$ `Purchase_Parts_A`
Hardware Producer Agent will have 13 units utility increase after the accomplishment of this schedule. If Hardware Producer Agent is half-cooperative to Computer Producer Agent, the best MQ schedule produced is the same as above. However if task Purchase_Parts_B comes with 7 units MQ_$ instead of 9 units, the best MQ schedule produced is as following:
$[10, 20]$ `Get_Hardware_A` $[20, 30]$ `Purchase_Parts_A`
Hardware Producer Agent will have 7 units utility increase after the accomplishment of this schedule.

A similar reasoning process also applies to the Transport Agent. The above example shows how an agent reacts in a negotiation process depends on its attitude towards the other agent regarding this issue, and also is affected by the other tasks on its agenda. The more cooperative an agent is, the more it will sacrifice its own utility for the other agent's utility increase. This integrative negotiation mechanism enables the agent to manage and reason about different cooperative attitudes it could have with another agent regarding a certain issue.

5. EXPERIMENTAL RESULTS

The example in Section 4 shows that an agent needs to sacrifice some of its own utility to be cooperative with another agent. The question is: Can cooperative agents improve social welfare*social welfare*[7]? Is it always true that a cooperative agent could improve the social welfare? When should an agent be cooperative and how cooperative it should be? To explore these questions, the following experimental work was done based on the scenario described in Section 4 where Hardware Producer Agent has a choice of three different attitudes toward Computer Producer Agent: completely-cooperative (C), half-cooperative (H), and self-interested (S), Transport Agent has the same three choices, so there are 9 combinations: SS (both agents are self-interested), SC (Hardware Producer Agent is self-interested while Transport Agent is completely-cooperative), SH (Hardware Producer Agent is self-interested while Transport Agent is half-cooperative), HS, HC, HH, CS, CH, CC. The data is collected over 48 groups of experiments; in each group of experiments, the agents work on the same incoming task set under the nine different situations. The tasks in each set for each group experiment are randomly generated with different rewards, deadlines and early reward rates within certain ranges.

Table 1 shows the comparison of each agent's utility and the social welfare under these different situations. The percentage numbers are the normalized utility numbers based on the utility gained when agent is self-interested. Table 1 shows that when both Hardware Producer Agent and Transport Agent are completely-cooperative to Computer Producer Agent (CC), the society gains the most social welfare. Even when both agent are only half-cooperative (HH), the social welfare is still very good. However, when one agent is completely-cooperative, the other agent is self-interested (CS, SC), the social welfare does not improve much compared to the completely self-interested (SS) case. The reason for the lack of significant improvement is that, in this example, to accomplish task "Produce_Computer" requires both task "Get_Hardware" and task "Deliver_Computer" to be successfully finished. When one agent is completely-cooperative, it sacrifices it own utility, but task "Produce_Computer" may still fail because the other agent is not cooperative, the utility of Computer Producer Agent does not increase as expected, and the global utility does not improve. This happens when the completion of a task is spread over more than two agents, the information from Computer Producer Agent about its utility increase is only an estimation, it depends not only on task "Get_Hardware" for Hardware Producer Agent, but also relies on task "Deliver_Computer" for Transport Agent. In this situation, if Hardware Producer Agent has no knowledge about the attitude of Transport Agent, it may not be a good idea to be completely-cooperative towards Computer Producer Agent. The above data also shows that the utility of Transport Agent does not decreases as much as Hardware Producer Agent when it becomes cooperative or half-cooperative, the reason is that, in the experimental set up, task "Deliver_Computer" takes less time than the task "Get_Hardware", so it is possible for Transport Agent to accept more tasks without losing too many high reward tasks from the outside.

Table 2 shows the statistical results about the difference between the social welfare under different cooperative situations using a t-test. For example, the first line in Table 2 shows that with the 0.01 Alpha-level, we can reject the hypothesis H_o that the difference between the social welfare when both agents are cooperative and the social welfare when both agents are self-interested is equal to 330[8], compared to the hypothesis H_a that that the difference between the social welfare when both agents are cooperative and the social welfare when both agents are self-interested is greater than 330.

Table 3 shows the expected utilities of Hardware Producer Agent and the expected social welfare under the three possible situations: when Hardware Producer Agent is self-interested, completely - cooperative and half-cooperative. When Hardware Producer Agent chooses one attitude, Transport Agent may adopt one of the three different attitudes. For example, when Hardware Producer Agent chooses to be self-interested, the global situation could be SS, SC, or SH. The utility numbers in the table in the expected values of the utilities under these three different situations. Table 3 tells us that when a cooperative task involves more than two agents and when the other agents' attitudes are unknown, being completely-cooperative means sacrificing its own utility significantly and thus is not a good idea. However, for this experiment setup, it is a good choice for an agent to be half-cooperative, sacrificing less of its own utility for more global utility increase. This is an example where the lack of a complete global view can be partially compensated for by having an agent acting in a partially cooperative attitude rather than being fully cooperative.

We recognized that the above conclusion may relate to the pa-

[7] Social welfare refers to the sum of the utilities of all the agent in the society which is considered, i.e. in above example, the social welfare is the sum of the utilities of the three agents: Computer Producer Agent, Hardware Producer Agent, and Transport Agent.

[8] 330 is 20% of social welfare under the SS situation(1649), and 180 is 11% of social welfare under the SS situation.

	Utility of Computer Producer Agent	Percentage	Utility of Hardware Producer Agent	Percentage	Utility of Transport Agent	Percentage	Social Welfare	Percentage
SS	218	1.000	575	1.000	856	1.000	1649	1.000
CC	842	4.08	415	0.72	766	0.90	2022	1.23
HH	587	2.84	493	0.86	806	0.94	1886	1.14
SC	301	1.41	587	1.02	798	0.93	1686	1.02
CS	469	2.24	364	0.63	839	0.98	1672	1.01
HS	390	1.87	467	0.81	845	0.99	1702	1.03
SH	292	1.36	585	1.02	815	0.95	1692	1.03
HC	632	3.06	500	0.87	772	0.90	1905	1.16
CH	761	3.68	405	0.70	802	0.94	1967	1.19

Table 1: Comparison of Performance

Object	Number to Compare	Ho	Ha	Result	Alpha	p
CC - SS	330	=330	> 330	Reject Ho	0.01	0.008
HH - SS	180	=180	>180	Reject Ho	0.01	0.0008
SC - SS	0	=0	>0	Fail to reject Ho	0.01	0.0179
CS - SS	0	=0	>0	Fail to reject Ho	0.01	0.0965

Table 2: Results from Statistical Tests

rameters of the experiments. Table 5 shows these parameters. For example, the third row of the table shows that Hardware Producer Agent receives two "Purchase_Parts" task every 15 time clicks, the reward for each Purchase_Parts falls in the range of [2, 10], and the duration of the task is 6. Every "Produce_Computer" task comes to Computer Producer Agent with a reward of 20 units of MQ_$, if it is finished by its deadline, Computer Producer Agent would have its local utility increased by 14 units (With the deduction of the 6 units of MQ_$ transferred to Hardware Producer Agent and Transport Agent). This information is sent to Hardware Producer Agent (and also Transport Agent) by attaching 7 (14 divided by 2 agents) units of *relational MQ* (MQ_*hc*/*t* for Hardware Producer Agent) with the task announcing proposal. This information is taken into consideration by the MQ scheduler when Hardware Producer Agent makes decision on this proposal. However, this information is not necessarily accurate because it is based on the assumption that the task "Produce_Computer" will be finished on time. Whether this assumption can become reality depends on whether Hardware Producer Agent and Transport Agent would accept the subcontract and fulfill them on time. The uncertainty associated with this information comes from the uncertainty of the other contractor agent's (Transport Agent) decision, where the other contractor agent's decision is based on the following issues:

1. the agent's attitude toward Computer Producer Agent (how cooperative it is); the more cooperative it is, the more likely this subcontract will be accepted.
2. the outside offers the agent receives: how good they are, how frequent they are and how they affect the subcontract task. If the outside offer is not good enough compared to the reward from the subcontract, or if they are not very frequent, or if they do not conflict with the subcontract task, the subcontract will be more likely to be accepted .

Because the above issues are unknown by Computer Producer Agent and Hardware Producer Agent, the uncertainty associated with the information about the local utility increase can not be resolved. This is why we make the statement at the beginning of this paper: **it is not possible from a computational or communicational perspective for an agent to be fully cooperative, because the agent needs to have complete global information to be fully cooperative.** Thus, it may be best for the organization to have agents being partially cooperative in their local negotiation with other agents rather than being fully cooperative in order to deal more effectively with the uncertainty of not having a more informed view of the state of the entire agent organization. Additional experiments have been done using different parameters. Table 4 shows the social welfare under different conditions. When the rewards of outside offers fall into the range of [11, 19], for the best of the social welfare, both agents should be self-interested.

Reward from outside offer	SS	CC	HH
[2, 10]	1.0	1.23	1.14
[11, 19]	1.0	0.93	0.98

Table 4: Social Welfare Using Different Parameters

Agent	Task	Reward	Frequence every 15 time clicks	Duration
c	Produce_Computer	20	1	16
h	Get_Hardware	3	1	7
h	Purchase_Parts	[2,10]	2	6
t	Deliver_Computer	3	1	6
t	Deliver_Product	[2,10]	2	7

Table 5: Experiment parameters

However, if there is no uncertainty or less uncertainty, it may be the best for the agent to be fully cooperative or more cooperative toward the group task in order to increase the social welfare. This does not mean the agent has to grant every subcontract of the group task, the decision also depends on the outside offer. If the outside offer is much much better than the subcontract even with the consideration of the contractee agent's utility increase, and if the contractor agent can only choose one between the subcontract of the group task and the outsider offer, the contractor agent will take the outside offer and drop the subcontract even if it is fully-cooperative. And in fact, this choice is good for the social welfare.

6. RELATED WORK

Glass and Grosz [4] developed a measure of social consciousness called "brownie points" (BP). The agent earns BP each time it chooses not to default a group task and loses BP when it does default for a better outside offer. The default of a group task may

	Utility of Hardware Producer Agent	Percentage	Social Welfare	Percentage
Self-Interested	583	1.0	1679	1
Completely-Cooperative	395	0.68	1887	1.13
Half-Cooperative	487	0.83	1831	1.09

Table 3: Utility of Hardware Producer Agent and Social Welfare

cause the agent to receive group tasks with less value in the future, hence reduces its long term utility. The agent counts BP as part of it overall utility beside the monetary utility. A parameter $BPweight$ can be adjusted to create agents with varying levels of social consciousness. This relates to our utility mapping function associated with the *relational MQ* which can be adjusted to reflect the agent's different attitude in negotiation. However, the *relational MQ* is agent-oriented and issue specific, so the agent can model different attitudes towards each agent and negotiation issue. Additionally, the mapping function can be a nonlinear function and describe a more complicated attitude. Their work assumes there is a central mechanism controlling the assignment of group tasks according to agent's rank (agent's previous default behavior), which is not always appropriated for an open agent environment. Instead, in our assumption, agents are all independent and there is no central control in the society. Axelrod's work [1] has shown stable cooperative behavior can arise when self-interesting agents adopt a reciprocating attitude toward each other. The agent cooperates with another agent who has cooperated with it in previous interactions. The idea of the reciprocity is related to our work if the *relational MQ* is used in bi-direction between agents, agent A collect some *relational MQ* from agent B and in the future the accumulated *relational MQ* could be used to ask agent B do some work for it, in this way, the *relational MQ* actually works as a quantitative measure of reciprocity. Sen developed a probabilistic reciprocity mechanism [7] in which the agent K chooses to help agent J with certain probability p and p is calculated based on the extra cost of this cooperation behavior and how much effort it owes agent J because agent J has helped it before. There are two parameters in the formula for calculating p which can be adjusted so that the agent can choose a specific cooperation level. However, this work assumes that cooperation always leads to aggregate gains for the group, and it was based on a known cost function - that is, they know how much extra it will cost then to do X for another agent. Neither of these two assumptions is necessary in our work. Also our work deals with more complex and realistic domains where tasks have real-time constraints and there are potentially complex interrelationship among tasks distributed across different agents. Other related work includes the cooperative negotiation work on task allocation[9], where the agents use the marginal utility gain and marginal utility cost to evaluate if it worth to accept a task contract in order to increase the global utility. However in this work, the agent acts as in a "completely-cooperative" mode and there is no choice on how cooperative it want to be.

7. CONCLUSION AND FUTURE WORK

We introduce an integrative negotiation mechanism which enables agents to interact over a spectrum of negotiation attitudes from self-interested to completely-cooperative in a uniform reasoning framework, namely the MQ framework. The agent can not only can choose to be self-interested or cooperative, but also choose how cooperative it wants to be. This provides the agent with a capability to dynamically adjust its negotiation attitude in a complex agent society. Experimental work shows it may not be a good idea to always be completely-cooperative in a situation involving an unknown agent's assistance; in that case, choosing to be half-cooperative may be good for both the individual agent and also for the society. Introducing this mechanism in the agent framework also strengthens the capability of multi-agent systems to model human societies. Multi-agent systems are important tools for developing and analyzing models and theories of interactivity in human societies. There are many complicated organizational relationships in human society, and every person plays a number of different roles and is involved in different organizations. A multi-agent system with this integrative negotiation mechanism is an ideal test-bed to model human society and to study negotiation and organization theories. In the future we plan to explore additional questions using this framework, such as: how should an agent choose it negotiation attitude based on its learning from past experience? How does different attitudes affects the agent's performance and the social welfare in different organizational contexts? and so forth.

8. REFERENCES

[1] *The Evolution of Cooperation*. Basic Books.

[2] *Negotiation*. Richard D. Irwin Inc, 1985.

[3] *Book on European perspectives on AMEC*, chapter Agents advanced features for negotiation in Electronic Commerce and Virtual Organisations formation process. Springer-Verlag, 2000.

[4] A. Glass and B. Grosz. Socially conscious decision-making.

[5] S. Lander and V. Lesser. Understanding the role of negotiation in distributed search among heterogeneous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[6] T. Sandholm and V. Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 126–133, 1996.

[7] S. Sen. Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Proc. of the Second International Conference on Multiagent Systems*.

[8] T. Wagner and V. Lesser. Evolving real-time local agent control for large-scale mas. In J. Meyer and M. Tambe, editors, *Intelligent Agents VIII (Proceedings of ATAL-01)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2002.

[9] X. Zhang, R. Podorozhny, and V. Lesser. Cooperative, multistep negotiation over a multi-dimensional utility function multi-agent systems negotiation. In *Proceedings of the IASTED International Conference, Artificial Intelligence and Soft Computing (ASC 2000)*.