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 introduces an integrative negotiation mechanism is introduced, which enables agents to choose any attitude from the extremes of self-interested and fully cooperative to those that are partially self-interested and partially cooperative. Experimental work verifies this mechanism and explores the question whether it always improves the social welfare to have an agent be completely cooperative. It is found that it is good 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.

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. 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 the following 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 [2]. 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. Secondly, 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. Thirdly, agents may be involved concurrently with more than one virtual organization while doing tasks for their own organizational entities. Additionally, 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 attitudes between these two extreme cases. 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

[*] 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 not withstanding 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.

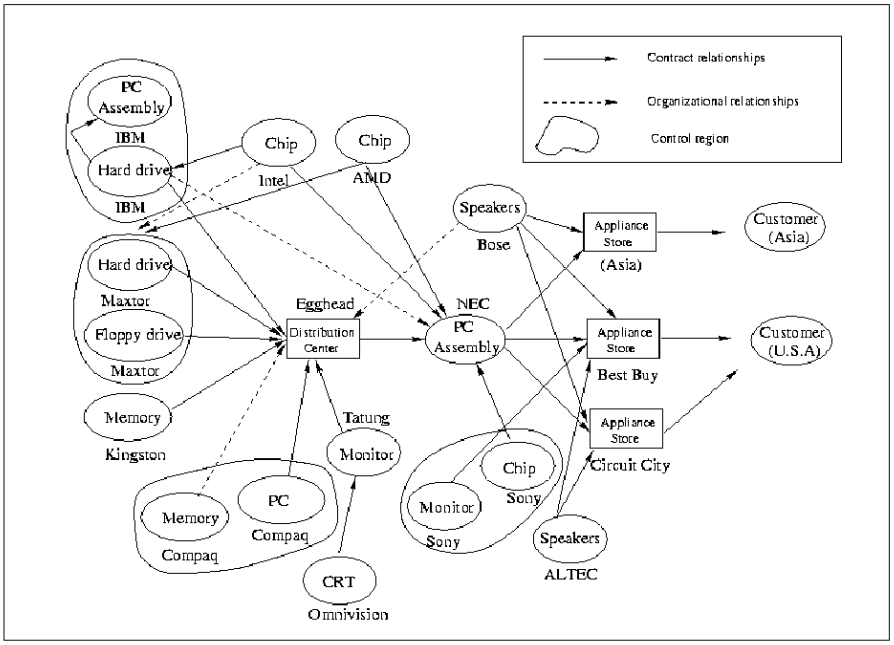

Figure 1. Supply chain example

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 attitudes. 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_2 to provide hard drives to agent_DIS rather. 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, 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 attitude dynamically.

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.

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 attitudes in the spectrum from purely self-interested to accommodative. It should be easy for the agent to switch from one attitude to another attitude.
2. The choice of negotiation attitude 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 that coordinates the agent's behavior. The agent should be free to choose any negotiation attitude according to its goals set by its designer.

So far, there has been no such negotiation framework that provides the above capabilities. So we introduce an integrative negotiation mechanism which enables agents to manage all sorts of negotiation attitudes in the spectrum from self-interested to completely cooperative in a uniform reasoning framework called the Motivational Quantities (MQ) framework [7], which is reviewed in Section 2. Section 3 describes this integrative negotiation mechanism. Section 4 uses examples to explain the ideas. Section 5 presents experimental results. Section 6 discusses related work and Section 7 concludes.

2 MQ Frameworks

The MQ framework [7] 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 cannot substitute for the progress toward another goal. Motivational Quantities (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. 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. MQ tasks are abstractions 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), deadline, process time needed to accomplishment this task (d_i), MQ production set ($MQPS$) that reflects the progress made by the accomplish of the task toward a goal, and MQ consumption set ($MQCS$) that 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, and 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.

3 Integrative Negotiation

In a complex agent society, an agent will need to work with other agents from a variety of different organizational positions. The agent's attitude toward negotiation is not just simply either competing or cooperative, the agent needs to qualitatively reason about each negotiation session, and so it can choose an appropriate negotiation attitude. Figure 2 describes this dual concern model [5]. 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 increase of the other agents' utility.

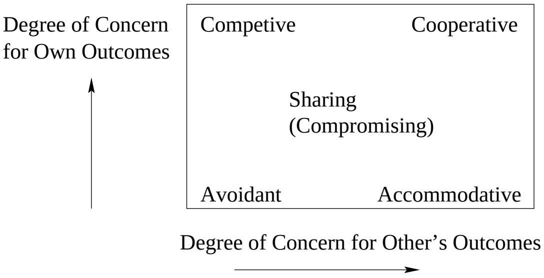
Degree of Concern for Own Outcomes
Competive
Cooperative
Sharing (Compromising)
Avoidant
Accommodative
Degree of Concern for Other's Outcomes

Figure 2. The dual concern model

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. For each task *t* allocated from agent A to agent B, certain MQs are transferred from agent A to agent B (an example of transferred MQ could be just currency). Agent B is motivated by the potential increase in its MQs to perform tasks for agent A. When B successfully accomplishes *t*, the agreed amount of the MQ will be transferred from agent A to agent B. There are two types of MQs that could be transferred with the successful accomplishment of task *t*: *goal_related MQ* and *relational MQ*. *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. 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*. 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). To reflect the B's attitude toward A's outcome[1], we intro-

[1] It is assumed that agents are honest and don't lie about the importance of task *t*. In this research we assume that agents are "inherently cooperative", they do not cheat or intent to hurt other agents even as they focus on their own goals. The social welfare or the performance of the organization can be a concern if the agent is built with such a goal. In the rest of this paper, we will use the word "self-interested" under this "inherently cooperative" assumption. "Self-interested" agent only considers its own

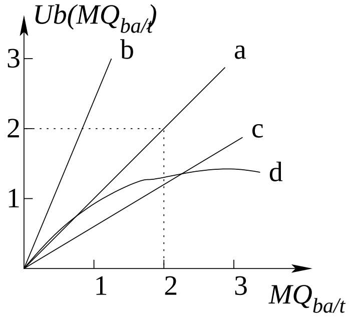

Figure 3. different mapping functions of $MQ_{ba/t}$

duce 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. The utility produced by a relational MQ can be seen as *virtual utility*, meaning utility that is not included in the social welfare computation. Actually, how $MQ_{ba/t}$ is mapped into agent B's virtual utility, 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 virtual utility.

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;

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.

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 its 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. Additionally, 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.

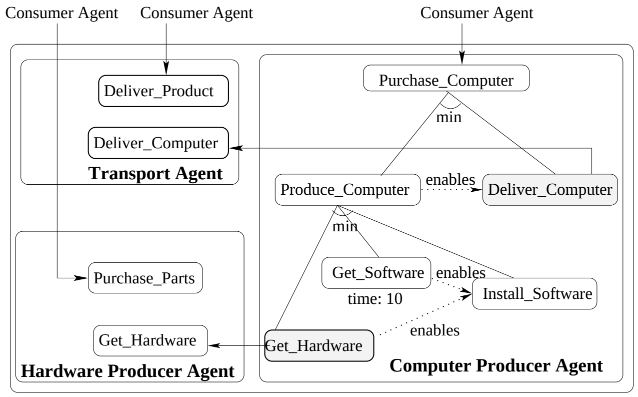

Figure 4. Agent Society

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. Computer Producer Agent (*c*) receives Produce_Computer task from an outside agent (which is not considered in this example). 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. Hardware Producer Agent (*h*) receives task Get_Hardware from Computer Producer Agent, it also receives Purchase_Parts task from an outside agent. 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:

- earliest start time (est), the performance of the task before this time does not generate valid results.
- 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 $MQ_\$$).

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_Computer task always goes to Transport Agent with a fixed reward of 3

outcome when it makes its local decision.

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_Computer 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_Computer. $MQ_{hc/t}$[2] 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. Suppose Computer Producer Agent receives the following task:

task name : Purchase_Computer_A
est: 10
deadline: 70
reward: 20 units $MQ_{\$}$

Through the reasoning of the MQ scheduler, Computer Producer Agent decides to accept it and finish it by time 40. Its local utility increases by 14 (after paying the sub-contractor agents) 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 $MQ_{\$}$* *7 units $MQ_{hc/t}$*	*3 units $MQ_{\$}$* *7 units $MQ_{tc/t}$*

In this example, we look at three different attitudes with a linear function: $U_{ha}(MQ_{hc/t}) = k * MQ_{hc/t}$, $k \in \{1, 0.5, 0\}$. 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 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	*est*	*deadline*	*process time*	*MQPS*
Get_Hardware_A	*10*	*20*	*10*	*[$MQ_{\$}$,3]* *[$MQ_{hc/t}$, 7]*
Purchase_Parts_A	*10*	*30*	*10*	*[$MQ_{\$}$,4]*
Purchase_Parts_B	*10*	*20*	*10*	*[$MQ_{\$}$,9]*

If Hardware Producer Agent is completely cooperative to Computer Producer Agent ($k = 1$), 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 ($k = 0$), 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 ($k = 0.5$), the best MQ schedule produced is the same as above. However, if task Purchase_Parts_B comes with 6 units $MQ_{\$}$ instead of 9 units, then 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.

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: Is it always true that a cooperative agent could improve the social welfare (Social welfare refers to the sum of the utilities of all the agent in the society which is considered.)? 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.

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. 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 agents are only half cooperative (HH), the social welfare is

[2] Similarly, $MQ_{tc/t}$ is a *relational MQ* that reflects the relationship of Transport Agent with Computer Producer Agent concerning task *t*.

	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

	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 2. Utility of Hardware Producer Agent and Social Welfare

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[3]. 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.

Table 2 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 is the expected values of the utilities under these three different situations. Table 2 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.

Agent	Task	Reward	Frequency	d
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 3. Experiment parameters

We recognized that the above conclusion might relate to the parameters of the experiments. Table 3 shows these parameters. 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 how cooperative the

[3] Results from t-test have shown that the difference of the social welfare between CC and SS, also between HH and SS, are statistically significant.

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 is toward Computer Producer Agent, and how good and how frequent the outside offers are. Because these issues are unknown by Computer Producer Agent and Hardware Producer Agent, the uncertainty associated with the information about the local utility increase cannot 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.

6 Related Work

Glass and Grosz [3] 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. A parameter $BPweight$ can be adjusted to create agents with varying levels of social consciousness. 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. Axelrod's work [1] has shown stable cooperative behavior can arise when self-interesting agents adopt a reciprocating attitude toward each other. 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 [6] 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 that 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. Our experimental work has shown that even in a cooperative system it may not be the best for the social welfare to have agents be fully cooperative. Similar result is also shown in [4], which uses the distributed constraint satisfaction model that is much different from the underlying model in this work.

7 Conclusion and Future Work

We introduce an integrative negotiation mechanism that 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 not only can choose to be self-interested or cooperative, but also can choose how cooperative it wants to be. This provides the agent with the capability to dynamically adjust its negotiation attitude in a complex agent society. Introducing this mechanism in the agent framework also strengthens the capability of multi-agent systems to model human societies. We recognize that the experimental result is not a general result, it does not answer the question that how cooperative an agent should be in a specific situation. We plan to develop an analytical model of the environment that enables the agent to predict the influence of its negotiation attitude on its own performance and also on the social welfare, hence to select the appropriate cooperative level to balance its own utility achievement and the social welfare.

References

[1] R. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.

[2] A. P. R. Eugenio Oliveira. *Book on European perspectives on AMEC*, chapter Agents advanced features for negotiation in Electronic Commerce and Virtual Organisations formation process. Springer-Verlag, June 2000.

[3] A. Glass and B. Grosz. Socially conscious decision-making. In *Proceedings of Agents 2000 Conference*, pages 217 – 224, Barcelona, Spain, June 2000.

[4] H. Jung, M. Tambe, and S. Kulkarni. Argumentation as distributed constraint satisfaction: applications and results. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 324–331, Montreal, Canada, May 2001. ACM Press.

[5] R. J. Lewicki and J. A. Litterer. *Negotiation*. Richard D. Irwin Inc, Homewood, Illinois, 1985.

[6] S. Sen. Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In *Proc. of the Second International Conference on Multiagent Systems*, pages 322–329, Menlo Park, CA, 1996. AAAI Press.

[7] 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.