How Cooperative Should I Be?*

Integrative Negotiation In Complex Organizational Agent Systems

Xiaoqin Zhang
Computer Science Department
University of Massachusetts at Amherst
shelley@cs.umass.edu

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

Tom Wagner
Automated Reasoning Group
Honeywell Laboratories
wagner_tom@htc.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. Categorized with a large grain-size, negotiation research falls into two general classes: cooperative negotiation and competitive negotiation. In competitive negotiation, agents are self-interested and they 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 [8, 9] 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 [2, 5, 11]. In this work, there is no notion of individual agent utility – agents are "completely-cooperative" with each other and cooperate to solve problems together.

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 maximize its local utility, nor distributed problem solving systems where all agents are completely-cooperative working to maximize the achievement of a set of global goals. This will occur for two reasons. One reason is that agents from different and separate organizational entities will come together to dynamically form virtual organization/team for solving specific problems that are relevant to each of their organizational entities[7]. How these agents work in their team will often dependent on the existence of both long term and short-term relationships and on the confrontational attitude of their underlying organizational entities. We also feel that 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 agent from other organizational entities. Additionally, agents may be involved concurrently with more than one virtual organizations while doing tasks for their own organizational entity. Secondly, we feel that even agents working solely with agents of their own organizational entity, it still may be advantageous for them to take varying attitudes in the spectrum of fully cooperative to totally self-interested in order for the organization to best achieve its overall goals. This perspective is based on a bounded-rational argument: it is not possible from a computational nor communication perspective for an agent to be fully cooperative, since agents need 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 is our feeling that it may be best for the organization to have agents being partially cooperative in its local negotiation with other agents rather than being fully cooperative in order to more effectively deal with uncertainty of not having a completely informed and up-to-date view of the state of the entire agent organization.

Multi-agent system will thus consist of large groups of loosely coupled agents that work together on tasks. The relationships among agents will depend on their organizational roles and may be of any

*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.

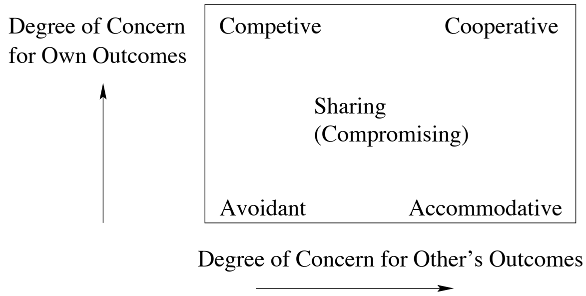

Figure 1: The dual concern model

type from purely self-interested to totally cooperative. This is the complex organizational problem space the MQ [12, 13] framework is designed to represent. Note that this work pertains to complex agents. We assume that *Agents* are autonomous, heterogeneous, persistent, computing entities that have the ability to choose which tasks to perform and when to perform them. Agents are also rationally bounded, resource bounded, and have limited knowledge of other agents.[1] Agents can perform tasks locally if they have sufficient resources and they may interact with other agents. The agents will have choices about with whom to collaborate, how to negotiate, what to charge for services, etc. Further, the negotiation strategy will be dependent on the relationships among the negotiating parties and the particular negotiation issue.

We therefore feel that 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 1 describes this dual concern model [6]. 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.

In this paper, we present an integrative mechanism that enables an agent to qualitatively manage its attitude towards each negotiation session. This mechanism is not purely self-interested or purely cooperative, but supports ranges of these behaviors so that the agent can reason about how cooperative it should be. This mechanism is based on the Motivational Quantities (MQ) framework [12, 13], which is introduced 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.

[1] As agents are heterogeneous, they may be associated with different corporate entities (privacy issues), and because the contextual valuation of tasks is generally an exponential problem we do not assume agents know each other's utility functions, plan libraries, etc.

2. MQ FRAMEWORKS

In the MQ framework, it is assumed that for an agent to perform a task, or to consider a task, the task must produce value for the local agent. On the surface, this implies that the MQ model is only for controlling interacting self-interested agents. This is not the case. The restriction is to guarantee the ability to compare tasks from a unified perspective.

Consider the issue of task value. When agents are isolated problem solving entities, task performance produces value that is entirely of local benefit. In Multi-Agent systems, value may be of local benefit and of benefit to other agents including the agent society as a whole. The extremes are also possible; tasks may be only of local benefit and tasks may be only of benefit to agents other than the local agent. This latter case appears problematic for the assumption above: *all tasks produce local value*. This case is problematic only on the surface. For the local agent to consider performing such a task, it must indeed have value, however, in this case the value is of a different type or class than the value of its other candidate tasks. The task, for example, may be performed to meet some organizational directive, e.g., *service requests from agent β*, or to reduce favors owed to the agent, to accumulate favors for future use with the agent, or because a different agent with which the local agent holds common goals requested it. In the MQ framework, all tasks have value or a *motivation* for performing the task where the value is determined both by the value of the task and by the importance of the organizational objective with which the task is associated (and the current state of goal achievement). This enables the agent to compare and value tasks that are associated with different organizational goals, or tasks that are detrimental to one organizational goal while having positive benefit to a different organizational goal, or tasks associated with different organizations entirely, or tasks motivated by self-interested reasons to cooperative reasons. The MQ framework quantifies these different underlying motivational factors and provides the means to compare them via a multi-attributed utility function. In the MQ framework:

- Each agent has a set of MQs or *motivational quantities* that it tracks and accumulates. MQs represent progress toward organizational goals. MQs are produced and consumed by task performance where the consumption or production properties are dependent on the context. For example, two agents interacting to achieve a shared organizational goal may both see an increase in the same local MQ levels as progress is made (this is not a zero sum game), whereas agents interacting to satisfy different goals may each obtain different types and quantities of MQs from the same interaction.
- Not all agents have the same MQ set. However, for two agents to form a commitment to a specific course of action, they must have at least one MQ in common (or have the means for forming an MQ dynamically). If they do not have an MQ in common, they lack any common goals or objectives and lack any common medium of exchange. (Proxy and reducibility are somewhat addressed in [12].)
- For each MQ_i belonging to an agent, it has a preference function or utility curve, U_{f_i}, that describes its preference for a particular quantity of the MQ, i.e., $\forall MQ_i,\ \exists U_{f_i}()$ such that $U_{f_i}(MQ_i) \mapsto U_i$ where U_i is the utility associated

with MQ_i and is not directly interchangeable with U_j unless $i = j$. Different agents may have different preferences for the same MQ_i. Preferences in the framework are defined by the relation between task performance and organizational goals or directives.

- An agent's overall utility at any given moment in time is a function of its different utilities: $U_{agent} = \gamma(U_i, U_j, U_k, ..)$. We make no assumptions about the properties of $\gamma()$, only that it enables agents to determine preference or dominance between two different agent states with respect to MQs.

MQ Tasks are abstractions of the primitive actions that the agent may carry out. MQ tasks:

- May have deadlines, $deadline_i$, for task performance beyond which performance of said task yields no useful results.
- May have earliest start times, $start_i$, for task performance before which performance of said task yields no useful results.
- Each MQ task consists of one or more MQ alternatives, where one alternative corresponds to a different performance profile of the task. In many ways, this extension simplifies reasoning with the preliminary model presented in [12] while at the same time increasing the representational power of the framework by coupling different durations with the other performance characteristics. Each alternative:
  - Requires some time or duration to execute, denoted d_i.
  - Produces some quantity of one or more MQs, called an *MQ production set* ($MQPS$), which is denoted by: $MQPS_{i,j,k} = \{q_i, q_j, q_k, ..\}$, where $\forall i,\ q_i \geq 0$. These quantities are *positive* and reflect the benefit derived from performing the task, e.g., progress toward a goal or the production of an artifact that can be exchanged with other agents. In this model, the two are equivalent.
  - Akin to the $MQPS$, tasks may also consume quantities of MQs. The specification of the MQs consumed by a task is called an *MQ consumption set* and denoted $MQCS_{i,j,k} = \{q_i, q_j, q_k, ..\}$, where $\forall i,\ q_i \leq 0$. Consumption sets model tasks consuming resources, or being detrimental to an organizational objective, or agents contracting work out to other agents, e.g., paying another agent to produce some desired result or another agent accumulating favors or good will as the result of task performance. Consumption sets are the *negative* side of task performance.
  - All quantities, e.g., $d_i, MQPS, MQCS$, are currently viewed from an expected value standpoint.
- $MQCS$ defines quantities that are required for task performance. If a task lacks sufficient MQs for execution it is deemed un-executable and will not be performed in any fashion. This means it will have zero duration, consume zero MQs, and will produce zero MQs.

Space limitations preclude a full presentation of the model, but it is sufficient for understanding how our integrative negotiation framework is built upon the MQ framework[2].

The MQ model can support comparison between tasks that are performed for different organizational motivations to task that are performed for other agents in return for financial gain to tasks that are performed for other agents for cooperative reasons. Via the different preferences for the different quantities, agent control can be modulated and agents can reason about mixtures of different task types and different motivations. The use of state in the model also facilitates contextually dependent behaviors or adjustments to behaviors over time. Agent α performing cooperative work with a closely allied agent, β, for instance, may need to balance this work with cooperative work with others over time. As α accumulates progress toward goals held in common with β (represented as an MQ), its preference may shift to the accumulation of other MQs. The use of utility for this application is flexible and very general and there are many different ways to relate organizational goal importance to the process of task valuation.

[2]This summary lacks definitions and properties necessary to actually build the framework and to use it in agents. This summary also lacks some of the motivations behind these design decisions. For more information, interested readers are advised to consult [12, 13].

3. INTEGRATIVE NEGOTIATION

The motivational qualities(MQ) [12, 13] framework provides an agent with the capability to reason about different goals in an open, dynamic and large-scale MAS, hence the agent can evaluate a negotiation issue from an organizational perspective. The MQ framework quantifies different underlying motivational factors and provides the means to compare them via a multi-attributed utility function. 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. In this work, we will assume that they do not change during the agent's life to simplify the experiments. 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. While agent B may actually have an organizational goal to accumulate MQs of this type[3], in this paper, for simplicity of presentation, we will assume that agent B does not have an organizational level goal to cooperate with agent A. Accordingly, 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[4] 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[5] 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, transferred to agent B, Figure 2 shows four different functions for mapping $MQ_{ba/t}$ to agent B's utility.

Function a, b and c are liner 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[6];

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 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.

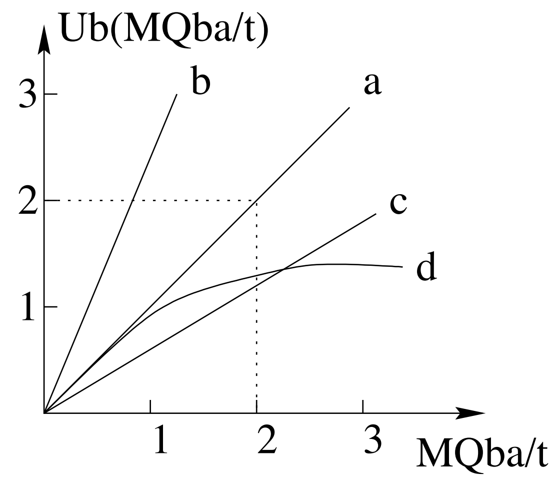

Figure 2: different mapping functions of $MQ_{ba/t}$

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 detailed 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 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 [10].

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

[3]In this case, the agent's local utility would also increase by accumulating MQs of this type, as an indication that cooperating with the other agent fosters its organizational objective

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

[5]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*

[6]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.

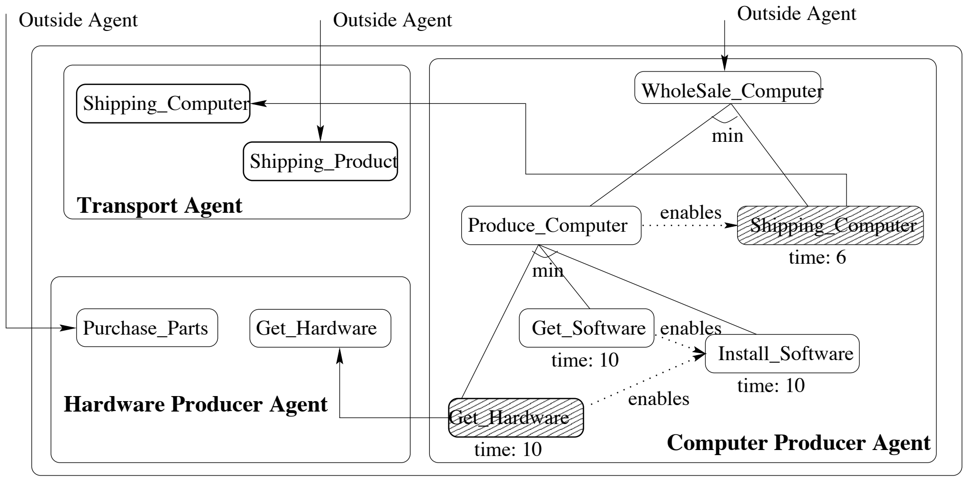

Figure 3: Agent Society

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.

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

4. THE SCENARIO

There are three agents in this society as shown in Figure 3:

1. Computer Producer Agent (*c*): receives "Produce_Computer" task from an outside agent (which is not considered in this example). Figure 3 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 ("Shipping_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 "Shipping_Computer" from Computer Producer Agent, it also receives "Shipping_Product" task from an outside agents.

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 "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)[7] in addition to the reward r.

Hardware Producer Agent receives "Purchase_Parts" task from an outside agent with x units MQ_$, x is a random number varying from 2 to 10. Computer Producer Agent has long-term contract relationship with Hardware Producer Agent and Transport Agent: its "Get_Hardware" task always goes to Hardware Producer Agent with a fixed reward of 3 units MQ_$, and its "Shipping_Product" task always goes to Transport Agent with a fixed reward of 3 units MQ_$. Every "Produce_Computer" task comes to Computer Producer Agent with a reward of 20 units MQ_$, if it is finished by its deadline, Computer Producer Agent would have its local utility increased by 14 units. Assume task "Get_Hardware" and "Shipping_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 $MQ_{hc/t}$ transferred with task "Get_Hardware" and 7 units $MQ_{tc/t}$ transferred with task "Shipping_Product". $MQ_{hc/t}$[8] 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
earliest start time: 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 follow two task requests are sent to Hardware Producer Agent and Transport Agent respectively:

task name : Get_Hardware_A
earliest start time: 10
deadline: 20

[7] 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.

[8] 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.

reward: 3 units MQ_$, 10 units "MQ_hc/t"
early finish reward rate: e=0.01[9]
task name : Shipping_Computer_A
earliest start time: 30
deadline: 40
reward: 3 units MQ_$, 10 units "MQ_tc/t"
early finish reward rate: e=0.01
In this example, we look at three different attitudes with a liner function: $U_{ha}(MQ_{hc/t}) = k * MQ_{hc/t}$.

1. k=1, Hardware Producer Agent is completely-cooperative to 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 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
earliest start time: 10
deadline: 20
process time: 10
MQPS: [MQ_$,3], [MQ_hc/t, 10]
task name : Purchase_Parts_A
earliest start time: 10
deadline: 30
process time: 10
MQPS: [MQ_$,4]
task name : Purchase_Parts_B
earliest start time: 10
deadline: 20
process time: 10
MQPS: [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 it 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.

[9]Assume Computer Producer Agent assigns the same early finish reward rate to this task as the task "Produce_Computer" it receives.

5. EXPERIMENT

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: Could cooperative agents make the social welfare[10] better? 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[11]. 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 "Shipping_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 "Shipping_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

[10]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.

[11]The experiments are performed in the MASS simulator environment[4], and the agents were built using the JAF agent framework[14]

	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

as much as Hardware Producer Agent when it becomes cooperative or half-cooperative, the reason is the following. In the experimental set up, task "Shipping_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 some statistical results about the difference between the social welfare under different cooperative situations using t-test. For example, the first line in Table 2 shows that with the 0.01 Alpha-level, we can accept the statement that the social welfare of the system when both agents are cooperative is at least 20% better than when both agents are self-interested[12].

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 number in the table in the expected value of the utilities under these three different situations. Table 4 shows similar information for Transport Agent. Table 3 tells us that when a cooperative operation 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, 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. For the Transport Agent which does not need to sacrifice too much to be completely-cooperative, it should always choose to be completely-cooperative.

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. The default of a group task may 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 [10] 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 owns 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 carry real-time constraints and there are potentially complex interrelationship among tasks distributed over different agents. Other related work includes the cooperative

[12] 330 is 20% of social welfare under SS situation(1649), 180 is 11% of social welfare under SS situation.

	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: the utility of Hardware Producer Agent and the social welfare

	Utility of Transport Agent	Percentage	Social Welfare	Percentage
Self-Interested	847	1.0	1675	1.0
Completely-Cooperative	803	0.95	1846	1.10
Half-Cooperative	818	0.97	1751	1.05

Table 4: the utility of Transport Agent and the social welfare

negotiation work on task allocation[15], 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 described in this paper 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 interact over a spectrum to manage of negotiation attitudes from self-interested to completely-cooperative in a uniform reasoning framework, namely the MQ framework. The agent not only can also choose to be self-interested or cooperative, but could choose how cooperative it wants to be. This provides the agent 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. 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] Axelrod, R. The Evolution of Cooperation. Basic Books.

[2] Conry, S.E., Kuwabara, K., Lesser, V.R. and Meyer, R.A. Multistage Negotiation in Distributed Constraint Satisfaction In IEEE Transactions on Systems, Man and Cybernetics, Volume 21, Number 6, pp. 1462-1477, November, 1992.

[3] Alyssa Glass and Barbara Grosz 2000. Socially Conscious Decision-Making. In the Proceedings of Agents2000 Conference, Barcelona, Spain, June 3-7, 2000. pp. 217 - 224.

[4] Horling, Bryan, Lesser, Victor, Vincent, Regis. Multi-Agent System Simulation Framework. In 16th IMACS World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation, EPFL, Lausanne, Switzerland, August 2000.

[5] Lander, S. and Lesser, V. Understanding the Role of Negotiation in Distributed Search Among Heterogeneous Agents. In Proceedings of the International Joint Conference on Artificial Intelligence, 1993.

[6] Roy J. Lewicki and Joseph A. Litterer Negotiation 1985, Richard D. Irwin, Inc. Homewood, Illinois.

[7] Eugenio Oliveira, Ana Paula Rocha. Agents advanced features for negotiation in Electronic Commerce and Virtual Organisations formation process In C. Sierra and F. Dignum, editors, Book on European perspectives on AMEC. Springer-Verlag, June 2000.

[8] Tuomas Sandholm and Victor Lesser Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. Proceedings of the First International Conference on Multi–Agent Systems, pp. 328-335, MIT Press, 1995.

[9] Sandholm, T. and Lesser, V. 1996. Advantages of a Leveled Commitment Contracting Protocol. Thirteenth National Conference on Artificial Intelligence (AAAI-96), pp. 126-133, Portland, OR, .

[10] Sandip 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, AAAI Press, Menlo Park, CA, 1996.

[11] Sandip Sen and Edmund H. Durfee A Formal Study of Distributed Meeting Scheduling Group Decision and Negotiation, volume 7, pages 265-289, 1998.

[12] Wagner, Thomas and Lesser, Victor. Relating Quantified Motivations for Organizationally Situated Agents. In Intelligent Agents VI: Agent Theories, Architectures, and Languages, Springer

[13] Thomas Wagner and Victor Lesser Evolving Real-Time Local Agent Control for Large-Scale MAS Intelligent Agents VIII (Proceedings of ATAL-01) editor,"J.J. Meyer and M. Tambe Springer-Verlag, Berlin, Lecture Notes in Artificial Intelligence, 2002

[14] Vincent, R.; Horling, B.; Lesser, V. An Agent Infrastructure to Build and Evaluate Multi-Agent Systems: The Java Agent Framework and Multi-Agent System Simulator. In Lecture Notes in Artificial Intelligence: Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems. Volume 1887, Wagner & Rana (eds.), Springer, pp. 102127, 2000

[15] Zhang, XiaoQin; Podorozhny, Rodion; Lesser, Victor. Cooperative, MultiStep Negotiation Over a Multi-Dimensional Utility Function. In Proceedings of the IASTED International Conference, Artificial Intelligence and Soft Computing (ASC 2000), 136-142, Banff, Canada, July, 2000, IASTED/ACTA Press.