A Proposed Approach to Sophisticated Negotiation*

Xiaoqin Zhang and Victor Lesser

December 7, 2000

1 Introduction

Negotiation, an interactive communication among participants to facilitate a distributed search process, is an important technique that used to effectively coordinate the behavior of agents in a Multi-Agent System(MAS). Negotiation is used for task allocation, resource allocation and conflict resolution.

As Muller[33] suggests, the research on negotiation can be classified into three categories: *negotiation language*, *negotiation decision* and *negotiation process*. The *negotiation language category* concerns the inter-agent part of negotiation such as the communication primitives for negotiation, their semantics and their usage in terms of a negotiation protocol. This category also includes the representation of the negotiation topics. The *negotiation decision category* deals with the intra-agent part of negotiation, such as algorithms to compare the negotiation topics and correlation functions for them. The definition of utility functions and the representation and the structure of the agents' preferences also belong to this category. Negotiation strategies fall in this category too. The *negotiation process category* is concerned with the global negotiation behavior of the individual agent and the negotiating society. General models of the negotiation process are investigated and the global behavior of the negotiation participants are analyzed in this category.

The work proposed for this research is motivated by following two questions. The first question is how should an agent deal with multiple negotiation issues when these issues are interconnected. The relationships among these negotiation issues can be classified as two types. One type of relationship is the *directly-linked* relationship: issue B affects issue A directly because issue B is a necessary resource (or a subtask) of issue A, the characteristics

---

*Effort 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 F30602-97-2-0032. 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.

(such as cost,finish time and quality) of issue B directly affect the characteristics of issue A. For example, in a supply chain problem, negotiations go on among more than two agents. The consumer agent negotiates with the producer agent and the producer agent needs to negotiate with the supplier agent. The negotiation between the produce agent and the supplier agent has a direct influence on the negotiation between the produce agent and the consumer agent. Another type of relationship is the *indirectly-linked* relationship: issue A relates to issue B because they compete for use of a common and limited resource. For example, agent A performs task T1 for agent B and also performs task T2 for agent C, because of the limited capability of agent A, when T1 is being performed affects when task T2 can be performed.

To our knowledge, there is no work that has addressed the *directly-linked* relationship in the negotiation process. There is some work on the *indirectly-linked* relationship among multiple negotiation issues. Level commitment[42] allows agent to decommit by paying a decommitment penalty. A statistical model is used to predict future events so that the agent can calculate the opportunity cost for the current commitment. When a new task arrives, the agent may retrack from its concurrent commitment by paying a decommit penalty to get an increasement on its local utility. In distributed meeting scheduling problem[49] [50], multiple meeting scheduling processes are going on concurrently. The agent can either block the proposed time or not block it until an agreement is reached, the commitment strategy affects the system's performance. Adaptive selection of the commitment strategy according to environment factors is recommended. However, in both of these work, the agent does not explicitly reason about the relationship among different issues under negotiation, so as to propose offers or counter-offers to minimize the conflict and optimize the combined outcome.

In general, multi-linked negotiation(including both the *direct-linked* and *indirect-linked* relationships) describes a situation where one agent needs to negotiate with multiple other agents about different issues, where the negotiation over one issue effects the negotiations over other issues. If these multiple negotiation processes are sequenced(which is obviously inefficient), the agent will have difficulty in evaluating a proposed commitment given that latter issues are undecided. However, if these processes are performed concurrently, the agent needs to manage them to avoid conflict and to find an acceptable combined solution for all issues under negotiation.

Another question we want to address in this research is negotiation in a complex organizational context. Until now all related work concerns either self-interested negotiation or cooperative negotiation. No work has been done to study negotiation between these two extreme cases. We feel that as the sophistication of Multi-Agent systems increases, they will be neither simple market systems where each agent is pure self-interested seeking to 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. Multi-Agent system will consist of large groups of loosely coupled agents that work together on tasks. The relationship between agents depends on their organizational role and could be any type of relationship ranging from pure self-interested to totally cooperative. The agents can choose to form a virtual organization[36] to work on a special common goal during a particular time period. The agents have choices about with whom to collaborate, how to negotiate, what to charge for services, etc. The negotiation strategy is dependent on the relationship between the negotiating parties and the particular negotiation issue.

To attack these two problems, we need to address our research work in all three categories of negotiation. In the *decision* category, we will use the motivational qualities(MQ)[56] framework to evaluate negotiation decisions. This framework provides the agent with the capability to reason about the relative importance of different goals and the values of achieving these goals based on organizational objectives. We will use this framework to support an integrated negotiation model that allowing agents to take different positions anywhere along the spectrum from the self-interested position to the cooperative position. As part of our use of this framework, we will extend it through the use of partial order scheduling technology . A partial order schedule representation is used as a reasoning tool to allow an agent to handle concurrent, multiple linked negotiation issues and evaluate the flexibility and the feasibility of the negotiation. In the *process* category, we propose that negotiation should be performed at different abstraction levels, rough commitments are formed at the upper level and then refined at the lower level to solve potential conflicts among different negotiation issues. In the *language* category, we will develop techniques to allow negotiation over multiple issues such as the temporal scope of the commitment, the cost of the commitment and the flexibility of the commitment.

2 Background Frameworks

We would like to first briefly describe the MQ (motivational quantities) framework[56] and the TÆMS framework[10] that are used as supporting architectures in this research. However, the major ideas are not restricted to these two frameworks, actually we feel they can also be applied to other suitable architectures. Though, we don't plan on demonstrating the latter point in this research.

2.1 MQ framework

In the MQ framework, the execution of a task contributes, in a quantitatively manner to the achievement of one or more agent's objectives. As part of this framework, there is a way of mapping this contribution to an overall utility increase associated with the potential execution of a task, given the agent's current state of achievement of different objectives. This enables the agent to compare 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 motivated by self-interested reasons to cooperative reasons. Each agent has a set of MQs or motivational quantities that it tracks and accumulates. MQs represent progress toward organizational goals and in certain cases may be used as a medium of exchange. MQs are produced and consumed by task performance where the consumption or production properties are dependent on the context. 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. Different agents may have different preference and organizational goals or directives.

MQ Tasks are abstractions of a patial ordered set of primitive actions that the agent may carry out. MQ tasks may have *deadlines* and *earliest start times.* Each MQ task consists of one or more MQ alternatives, where each alternative corresponds to a different performance profile of the task. Each alternative requires some time or *duration* to execute, produces some quantity of one or more MQs, called *MQ production set*(MQPS), and consumes some quantity of MQs, called *MQ consumption set*(MQCS).

2.2 TÆMS framework

The TÆMS task modeling language[10] (See Figure 3) is a domain-independent framework used to model the agent's candidate activities. It is a hierarchical task representation language that features the ability to express alternative ways of performing tasks, statistical characterization of methods(primitive tasks) via discrete probability distributions in three dimensions (quality, cost and duration), explicit representation of interactions between tasks, and resource requirements of methods. Quality is a deliberately abstract domain-independent concept that describes the contribution of a particular method to overall problem solving. Thus, different applications have different notions of what corresponds to model quality. Duration describes the amount of time that the method will take to execute and cost describes the financial or opportunity cost inherent in performing this action.

Hard and soft interactions between tasks, called NLEs(non-local effects), are also represented in TÆMS and reasoned about during scheduling and negotiation. Hard task interactions delineate hard precedence constraints such as *enables* and *disables.* Soft task interactions denote situations where result of one activity can *facilitate* or *hinder* another activity. Task resource consumption and production behaviors are modeled in TÆMS via *consumes* and *produces* task/resource nles - these nles describe the quantity of resources consumed or produced by task execution.

3 Major Ideas

3.1 Basic Assumptions

Before we introduce our negotiation framework, we would like to clarify our assumptions about the agent and their relationships in a Multi-Agent system.

1. Agents are autonomous, heterogenous, persistent, computing entities that have ability to choose which tasks to perform and when to perform them. Agents are also rationally bounded, resource bounded, and have limited knowledge of the state of other agents.

2. Agents are utility-driven and motivated to maximize the local utility given the limited resources. An agent's overall utility at any given moment in time is a function of its

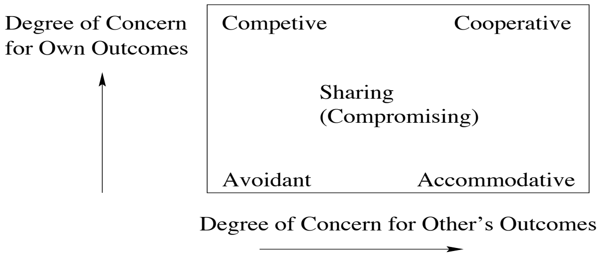

Figure 1: The dual concern model

different utilities. Each MQ can be mapped into a different type of utility according to a preference function or utility curve. The value of a certain amount of certain type MQ at a given time moment is related to the agent's current MQ state.

3. MQs represent progress towards organizational goals and in certain cases are used as a medium of exchange. MQs are produced and consumed by task performance. When a task is relocated from one agent to another agent, there are certain MQs transferred with the task. The type and the amount of the transferred MQ depends on the relationship among these agents.

4. When agents are interacting to achieve a shared organizational goal, each agent gets a local utility increase. Agents can negotiate to decide what approach each should choose to accomplish their subtasks and how they will split the achieved MQs among them.

5. A MQ task may have multiple MQ production sets, that means a given task may produce different groups of MQs. For instance, the execution of task T1 of agent A1 may contribute to one of its local goals and at the same time facilitate the execution of task T2 that belongs to agent A2. If agent A1 can perform task T1 with certain temporal constraints so as to facilitate task T2, it will get additional MQs from agent T2.

6. Although agents are motivated to maximize their local utilities, but they are not necessarily self-interested. Agent A could be cooperative to agent B if agent A counts the contribution to the increase of agent B's utility as a special MQ_{ba}that contributes to its utility. $MQ_{ba}$is one type of MQ that motivates agent A to cooperate with agent B to increase their combined utilities. This interaction style knowledge of with which agents to be cooperative and the degree of cooperative is part of the organizational knowledge of the agent; it could be provided by the agent designer or accumulated through learning based on the runtime experience[49].

3.2 Integrative Negotiation

There are two general types of negotiation that are studied: cooperative negotiation and competitive negotiation. In a competitive negotiation, agents are self-interested, they negotiate to maximize their own local utility; in a cooperative negotiation, agents are working to find a solution that increases their joint utility - the sum of the utilities of all involved agents. Between these two extreme situations, there are potentially many other options. These other options depend on the agent's attitude towards the importance it attaches to the increase of its own utility verse the importance it attaches to help other agents increase their utilities.

Figure 1 [28] describes this dual concern model. When the agent attaches importance to its own outcome, its attitude towards the negotiation is competitive(self-interested); when the agent attaches the same degree of importance to its own outcome as that of the other agent's outcome, its attitude is cooperative; when the agent attaches importance to other agent's outcome and no importance to its own outcome, its attitude is accommodative; if the agent attaches no importance to either outcome, its attitude is avoidant(the negotiation is not worth its time and effort).

In a complex agent society, the agent needs to work with agents from different organizational positions, such as an agent from its own group, an agent from a higher position in its company, an agent from a cooperative company, or an agent from a competing company and so forth. The agent's attitude toward a negotiation issue is not just simply either competing or cooperative, the agent needs to quantitatively reason about each negotiation session- how important its own outcome is as related to the other agent's outcome, so it can choose an appropriate negotiation strategy.

How can an agent choose the negotiation strategy in such a complex organization context? One approach is to embed related information (i.e. "with agent A use strategy No.1") as part of the organizational knowledge. One shortcoming of this approach is that agents have difficulties when there is a new agent joining this society. Moreover, this "agent/strategy" type of knowledge could not be "issue-specific"; given an agent could play multiple roles, there could be different issues negotiated between agents, and the agents should select different strategy 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 to for a ride, even both requests come from the same agent, the negotiation strategy could be different. Another approach is that the agent dynamically selects the negotiation strategy by analyzing the other party, the issue in negotiation and its current problem-solving status. The following information could contribute to the selection of the negotiation strategy: "Who is the other agent?", "What are its reputation and style?", "What is its objective?", "How is its relationship to me?", "Are there other competitors?" and so forth. Some of this information can be learned from experience.

The motivational qualities(MQ)[56] 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 perspectives. The MQ framework quantifies different underlying motivational factors and provides the means to compare them via a multi-attributed utility function. The agent's attitude towards a negotiation issue is affected by the utility mapping function of the transferred MQ with this issue, which reflects the agent's attitude toward the other agent's outcome. We introduce a special MQ called $MQ_{ba/nl}$, which represents how cooperative agent A is to agent B on the concern

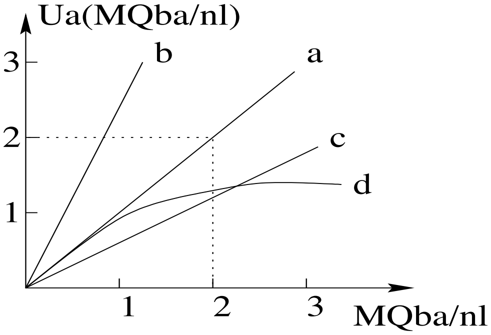

Figure 2: different mapping functions of MQba

of a special issue - task NL. Let $MQ_{ba/nl}$ be the type of MQ transferred from agent B to agent A when agent A performs task NL for agent B. $MQ_{ba/nl}$ is one MQ, that can not be transferred to any other kind MQ, and it can not be transferred to any other agents; in another word, $MQ_{ba/nl}$ is useless for agent A in the market, it is simply to measure the relationship between agent A and B. Actually, how $MQ_{ba/nl}$ is mapped into agent A's utility depends on how cooperative agent A is to agent B. Suppose there is certain amount $MQ_{ba/nl}$, representing the utility agent B gained by having agent A perform task NL, transferred to agent A, Figure 2 shows four different functions for mapping $MQ_{ba/nl}$ to agent A's utility.

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

If $k = 1$(a), $U_a(MQ_{ba/nl}) = MQ_{ba/nl} = U_b(NL)$ ($U_b(NL)$ denotes the utility agent B gained by transferring NL), then agent A is totally cooperative to agent B;

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

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

If $k = 0$, $U_a(MQ_{ba/nl}) = 0$, the agent A is self-interested with respect to agent B; in this case, if agent B wants agent A to do NL, it needs to pay another kind of MQ to agent A.

The mapping function could be a nonlinear function(d) that describes a more complicate attitude of agent A to agent B.

We will use this utility analysis framework to support the development of a family of negotiation protocols that allows a range of agent relationships to be accommodated in the negotiation process.

3.3 Multi-Leveled Negotiation

Usually negotiation is structured as a single level process: from the proposal to the final commitment, all related issues such as finishing time, achieved quality and offered price are determined in this process. Given the uncertainty of task execution and several other related issues, it is difficult to construct an integrated framework in which all these issues are addressed concurrently and done so in an efficient way. So we propose a multi-leveled negotiation framework in which the negotiation process is performed at different abstraction levels. The upper level deals with the formation of high level goals and objectives for the agent, and the decision about whether or not to negotiate with other agents to achieve particular goals or bring about particular objectives. The negotiation at this upper level determines the rough scope of the commitment (i.e. the time and the quality characteristics) and the cost of the commitment. The lower level deals with feasibility and implementation operations, such as the detailed analysis of candidate tasks and actions and the formation of the detailed temporal/resource-specific commitments among agents. The negotiation at this lower level involves the refinement of the rough commitments from the upper level.

It is reasonable for an agent to evaluate the importance of a commitment from the upper level. An agent has a better understanding of how a commitment could affect its local plan hence its utility gain when it reasons about this commitment in the upper level framework. Moreover, the agent needs some initial commitments when it chooses its local plan. For example, suppose agent A needs to perform task T and there are three available plans, each one has different quality, duration and cost characteristics. The plan P3 requests assigning a subtask Mc to another agent. From the high level view, if the agent A can find another agent to perform the subtask Mc before time 20 and with transferred utility less than 5, the plan P3 is the best choice. If such a commitment is not available, the agent A needs to choose another plan for task T.

On the other hand, not all issues could be modeled or totally decided on the upper level. The upper level deals with the agent's high level activity plan, it lacks detailed information of each activities hence it is difficult to reason about the agent's detailed activities. There are two kinds of issues related to the decision-making process in the negotiation. Those issues - which have strong influence on local plan selection and involve utility transferred between agents (i.e. an important non-local task or an important resource that needs to be purchased from other agent), should be negotiated first at the upper level and rough commitments should be constructed for them. However, those issues, which have less influence on local plan selection and involve reasoning about the detailed structure of the low level activities, can not be modeled on the upper level and do not need to be decided on the upper level. These issues include:

1. *Internal relationships between subtasks that belongs to different high level tasks.* For instance, the subtask "*go to pharmacy*" that belongs to "*take care of sick sister*" facilitates the subtask "*go to post office*" that belongs to "*send gift to a friend*" because the pharmacy is next to the post office. This relationship is not visible from the high level tasks, but the agent can exploit it to optimize its local plan after the high level plan is decided.

2. *Uncertainty of the execution characteristics that are not visible on the higher level.* The agent is uncertain about the task's duration, cost and quality produced when it makes plan about the task. Expected values(or other abstraction model, such

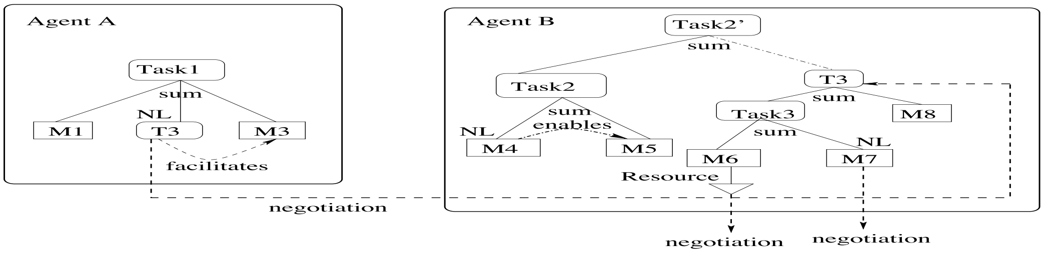

Figure 3: Agent A has a nonlocal task T3 contracted to agent B while agent B needs to subcontract M7(a subtask of T3) to another agent and request resource for M6(another subtask for T3) through negotiation.

as a range with certain confidence level, see Section 4.9) are used in the high level planning and uncertainties are not taken into account. This leads to more efficient processing at the higher level, however, in certain situation detailed reasoning about uncertainty becomes important to making commitment. The lower level has detailed information about the uncertainty, and as more context knowledge is available along with the process, thus the high level commitment can be adjusted to accommodate for uncertainty.

3. *Internal resource requirement associated with low level tasks.* For example, there is an agent who shares a printer with several other agents, given the knowledge of the general printing load, the agent knows it is unnecessary to reserve the printer when it builds its high level plan. But when the agent comes to arrange its local activities, it should taken this resource constraint into consideration.

Considering the above issues, the agent needs to revise high level commitments through low level negotiation and reorder its local level activities, hence to optimize it local plan and commitments , reduce failure possibilities, avoid conflicts and achieve higher utilities.

The basic idea of multi-level negotiation is described as follows. First agents negotiate on the upper level to build rough commitments, which are the foundation for other planning and coordination/negotiation activities. As a result of these activities, more constraints are added, and more detailed information is available from the lower level; these sketchy commitments then are refined as a result of this new information and these new constraints at the lower level.

The related issues need to be studied here include: What issues should be negotiated on which level? How does the negotiation on one level affect the negotiation on the other level? How does the agent control the multi-leveled negotiation?

3.4 Multi-Linked Negotiation

In the multi-task, resource sharing environment, an agent needs to deal with multiple related negotiation issues including:

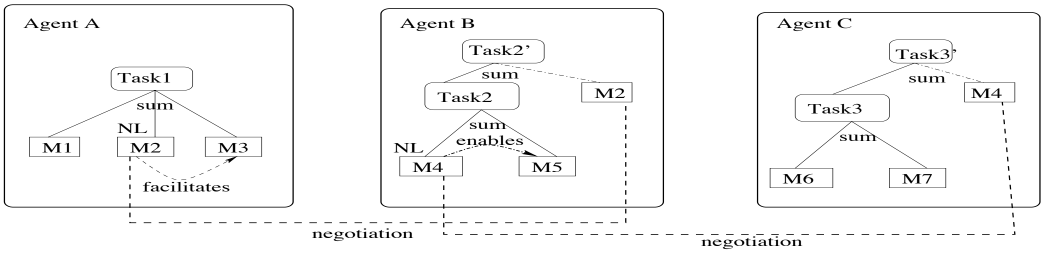

Figure 4: Agent A has a nonlocal task M2 contracted to agent B while agent B has a nonlocal task M4 contracted to agent C.

1. task contracted to other agents;
2. task requested by other agents;
3. external resource requirement for local activities;
4. interrelationship among activities distributed on different agents ;

These issues are related to each other. The result of one issue has influence on other issues. An example of a complex situation is the negotiation chain problem. Agent A has an issue x negotiated with agent B and agent B has an issue y negotiated with agent C. The negotiation between agent B agent C over issue y affects the negotiation between agent A and B over issue x. As we described in section 1, the relationships among related negotiation issues could be classified as *directly-linked* relationships and *indirectly-linked* relationships. Figure 3 describes a *directly-linked* relationship, Figure 4 describes a *indirectly-linked* relationship, and Figure 5 describes a situation between *directly-linked* and *indirectly-linked* relationship. If the *facilitates* relationship is exploited, the negotiation on M2 and the negotiation on M4 are *directly-linked*; otherwise, they are *indirectly-linked*.

How can the agent deal with these multiple related negotiation issues? One solution is to deal with these issues in sequence. The drawback is inefficiency and the difficulty of finding a good solution from a global perspective. For example, in Figure 5, agent A has two non-local tasks, task M2 contracted to agent B and task M4 contracted to agent C. If M2 could be finished before M4 starts, it will facilitate the performing of M4. Suppose agent A first negotiates with agent B and then negotiate with agent C, it tries to push task M2 to be finished as soon as possible so M2 can facilitate M4. Through the negotiation with agent B, it is decided that M2 is finished by time 10 (agent A pays a high cost for this commitment), but then it is found that task M4 can not be started before time 20 because of the other local activities of agent C. Given this latter information, it is not worth paying a high cost for M2 to be finished by time 10. This example shows a shortcoming of the sequential negotiation.

Concurrent negotiation is another choice. Multi-Leveled framework is suitable for concurrent negotiation. When an agent performs the high level concurrent negotiation, it tries to minimize the possibility of conflicts among different negotiation issues. When the agent has more detailed information, it can solve the conflicts through the low-level negotiation.

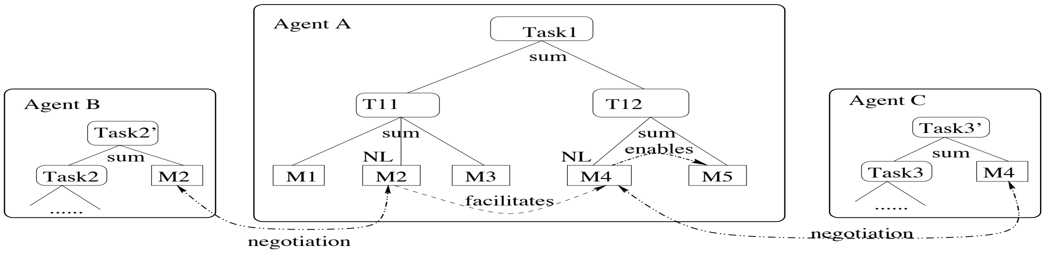

Figure 5: Agent A has two nonlocal tasks: M2 contracted to agent B and M4 contracted to agent C.

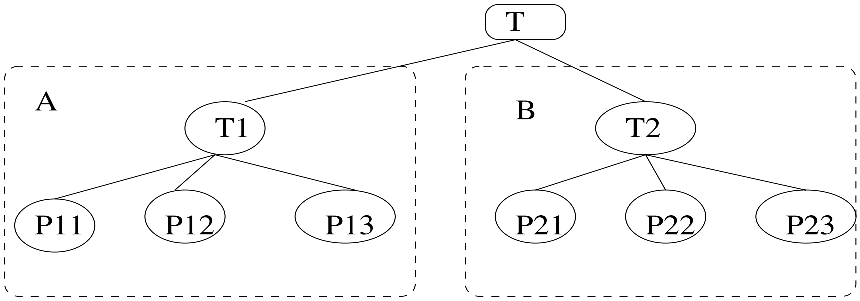

Figure 6: joint goal model

A partial order scheduler (see Section 4.5) will be used as a basic reasoning tool for concurrent multi-linked negotiation. It can be used to reason about the influence of a commitment of one issue on other negotiating issues. It also can be used to reason about the flexibility of each commitment and how it affects the flexibility of its local activities.

3.5 Joint Goal

In section 2, we describe the task allocation process in a contracting model, one agent contracts one of its subtask to another agent. Another coordination model of sharing task among agents is the joint goal model. As shown in Figure 6, agent A and agent B have a joint goal T, agent A has the subtask T1, and agent B has subtask the T2. Each agent has alternative plans to perform its subtask, and each plan has different quality, completion time, and utility cost characteristics. These alternative plans are constructed by the DTC scheduler, and the utility cost is evaluated by the MQ level scheduler as described in section 2. The combined quality and completion time of T1 and T2 determine the quality and completion time of task T, which can be mapped into the MQPS(T) - the MQ agent A and agent B gain for performing task T.

Suppose agent A and B are cooperative, they would like to do a cooperative search to find such a combined plan (P1i, P2j) to maximize the combined utility gain:

MUG(T[P1i, P2j]) - MUC(P1i) - MUC(P2j)

The MUG(T[P1i, P2j]) is not a fixed number , it depends on how agent A and B share the MQ - MQPS(T[P1i, P2j]), and the function that maps the MQ to the agent's utility

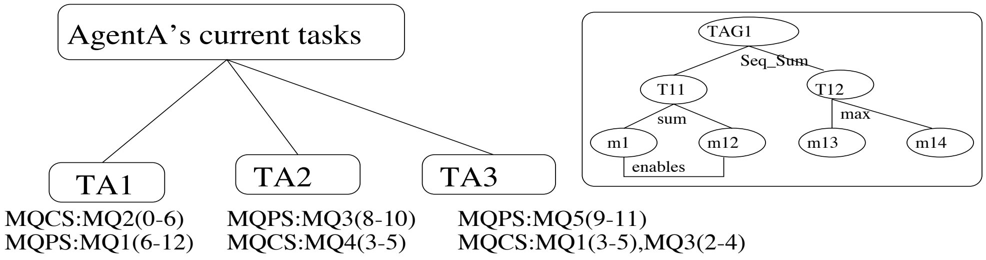

Figure 7: An example of MQ tasks

(agent A and B may have different mapping functions) .

For example, MQPS(T[P1i, P2j]) = {mq1: 10 units; mq2: 8 units}.

Agent A: utility(mq1)= 1.2*amount(mq1); utility(mq2)= 0.6*amount(mq2);

Agent B: utility(mq1)= 0.5*amount(mq1); utility(mq2)= 0.9*amount(mq2);

The best sharing scheme is agent A get all 10 units mq1 and agent B get all 8 units mq2, this scheme maximizes their combined utility gain as: 1.2*10 + 0.9*8 =19.2. Actually the mapping function can be more complex than this simple linear function in this example, however, there exists a sharing scheme that maximizes their combined utility.

The question here is not only to maximize the combine utility: MUG(T[P1i, P2j]), it is to maximize the combined utility gain: MUG(T[P1i, P2j]) - MUC(P1i) - MUC(P2j). It needs a two dimensional search along both the combined plan the sharing scheme.

4 Approach

In this section, examples will be developed to explain in detail how the multi-leveled, multi-linked integrative negotiation works. Additionally, the interesting intellectual problems and the proposed ideas to solve them will be presented.

4.1 Overview of Basic Ideas

The MQ model[56] describes the agent’s organization knowledge about task utility but it lacks a detailed model of tasks and their interactions, the uncertainty characteristics and resource requirements of tasks, which belong to the TÆMS[10] model. The proper integration of these technologies can give agents the benefits of both reasoning about the organizational concerns and handling detailed feasibility analysis and implementation of objectives.

An agent has its MQ level view of its local activities, that is a set of potential MQ tasks, each has been associated with certain MQPS(the type and amount of MQ this task produces) and MQCS(the type and amount of MQ this task consumes), which can be mapped into the agent’s utility given the agent’s current MQ state. For example, Figure 7 shows agent A has three MQ tasks, TA1, TA2 and TA3. TA1 produces MQ1 from 6 units to 12 units, and it consumes MQ2 from 0 units to 6 units. The amount of the MQ varies

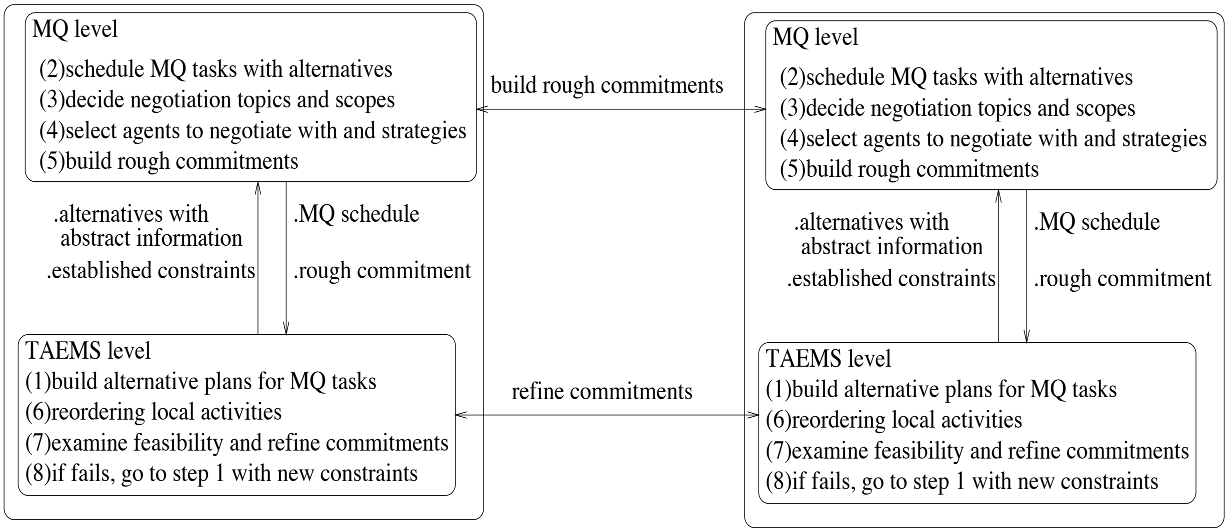

Figure 8: two level negotiation framework

depending on what plan is used to accomplish task TA1. For each MQ task T, there is a TÆMS task group TG that describes the detailed activities for this task, i.e. the task group TAG1 describes the detailed activities in task TA1. Different plans to accomplish the MQ task T can be generated from the TÆMS task group TG by the DTC scheduler, each plan has different quality, duration and cost characteristics that affect the MQPS and MQCS of the task T. This is the first step [step 1] shown in Figure 8 , which describes the two-level negotiation framework.

The extended MQ scheduler generates a partial order schedule that indicates what tasks the agent should attempt to execute, what plans are used to execute these tasks, and the execution ordering. This schedule represents the agent's best choice about what activities it should do maximize its local utility increase [step 2]. Based on these schedules, the agent can reason about the utility of a specific commitment(i.e. contracting a task out to another agent, performing a task for another agent, or receiving external resource needed by one of its tasks). The negotiation on MQ level is a multi-dimensional negotiation that includes the amount of the transferred MQ, the temporal constraints of the commitment and the quality constraints of the commitment [step3]. Also the agent could select which agents to negotiate with and the appropriate negotiation strategy according to organizational relationships and the negotiation issues[step 4]. A partial order schedule makes it possible for the agent to reason about how a commitment affects the flexibility to modify the execution constraints on other local activities and the relationships among multiple related negotiation issues. The partial order schedule on MQ level has the MQ task as the basic reasoning element. The MQ level negotiation builds rough(partial-specified) commitments for those issues that should or could be reasoned on the MQ level[step 5].

After building a local MQ schedule and rough commitments on MQ level, the agent reorders its local activities on the TÆMS level[step 6]. Low level relationships among TÆMS tasks/methods and detailed resource constraints are taken into account in this

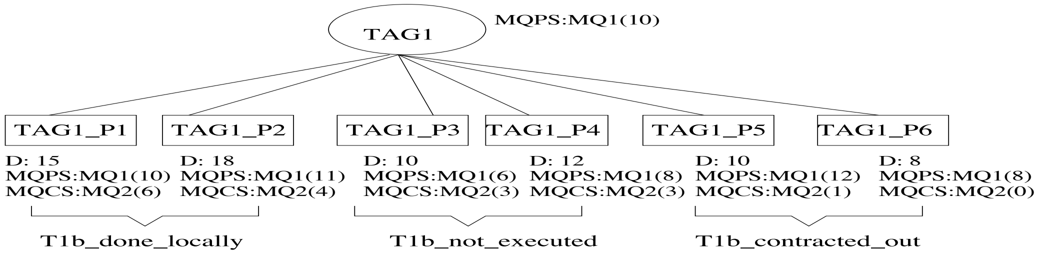

Figure 9: TAG1's alternatives

reordering process. In this reordering process, the agent could optimize its local schedule by taking advantage of the interrelationship among low-level tasks/methods, also the agent can verify the feasibility of its local schedule given rough commitments from the MQ level and those additional constraints from the TÆMS level[step 7]. A partial order schedule is also used to manage and reason about these relationships and constraints on the TÆMS level. The TÆMS level partial order schedule takes the TÆMS method as the basic reasoning element. Negotiation on the TÆMS level involves refining of those rough commitments as needed when:

1. there are conflicts or potential conflicts among commitments and local activities;
2. it is possible to reduce local cost or increase local utility by refining a commitment;

If the agent could find a feasible local schedule by reordering and re-negotiation on the TÆMS level, the agent can execute its local schedule and perform all of its commitments; otherwise, if conflict can not be resolved given all constraints, the agent needs to discard some commitments, establish other commitments on already scheduled local activities and go back to the MQ level to reschedule and may need to build some new commitments[step 8].

We will discuss more details about this two-level negotiation in following sections.

4.2 DTC scheduler builds alternatives

Design-To-Criteria(DTC) scheduler[55] is a domain-independent scheduler that aims to find a feasible schedule that matches the agent's particular criteria request. In this research work, it will be used off-line to build a library of alternative plans for achievement of a TÆMS task group. For example, agent A has three MQ level tasks TA1, TA2 and TA3, which are mapped into the task groups TAG1, TAG2 and TAG3 in the TÆMS model. Suppose there is a subtask T1b of TAG1 that potentially can be contracted to another agent.

DTC scheduler works on TAG1 according to following different assumptions: T1b is executed locally; T1b is not executed; T1b is contracted to another agent. These assumptions can be combined with different q, c, d scheduling criteria to generate several alternative plans as shown in Figure 9. Each plan has different q, c, d characteristics, corresponding

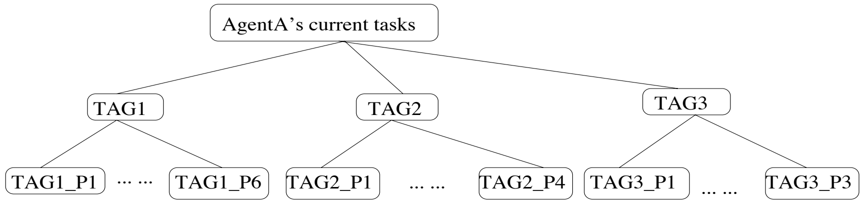

Figure 10: MQ level tasks

to a MQ level alternative with different duration, MQPS, and MQCS. For those plans that need to contract T1b to another agent, such as TAG1_P5 and TAG1_P6, the MQCS does not include the cost for contracting the task T1b , because the cost is unknown at this time. Similarly, different plans are generated for task TA2 and TA3.

Some options are not available in certain situation. For example, if the task T1b is necessary for the completion of task TA1, then the option *T1b_not_ executed* is not available; if agent A lacks the ability to perform task T1b, the option *T1b_done_locally* is not available. As part of this off-line process, there is a procedure associated with each task group that will analyze each plan to generate the appropriate MQ set for that plan.

This abstraction process can be done off-line, and these alternative plans can be stored in the agent's database. Not all alternatives are used in the MQ level scheduling process, a set of plans are selected according to current problem-solving context. For example, if the current minimum quality request for the task is 10, then those plans with achieved quality less than 10 are discarded and not used by the MQ scheduler.

4.3 MQ level scheduling

MQ level scheduler schedules these alternatives for TA1, TA2 and TA3 to find the best schedule MQ_S1 that provides the agent the most utility increase from its current state (Figure 10). If the plan TAG1_P5 or TAG1_P6 (T1b is contracted out) appears in the scheduler MQ_S1, agent A need consider contracting T1b to another agent; otherwise, agent A may choose to execute T1b locally or not to perform T1b as the schedule MQ_S1 recommends. Suppose the best schedule MQ_S1 includes the TAG1_P5 plan:

TAG1_P5[duration:10 earliest_start_time:0 deadline:20]
TAG2_P2[duration:10 earliest_start_time:0 deadline:30]
TAG3_P1[duration:15 earliest_start_time:10 deadline:40]

This is a partial order schedule, TAG1_P5 and TAG2_P2 need to be finished before TAG3_P1 starts. The reason is that TAG3_P1 consumes the MQs produced by TAG1_P5 and TAG2_P2. This partial order schedule can be expressed graphically as shown in Figure 11. Agent A compares the utility of the best schedule including the contracting plan of T1b(MQ_S1) with the utility of the best schedule without the contracting plan of T1b(MQ_S2), the difference is the utility gained by contracting T1b to other agent.

Marginal_Utility_Gain(T1b) = Utility(MQ_S1) - Utility(MQ_S2)

The basic constraint of the quality request and the temporal constraint of T1b is es-

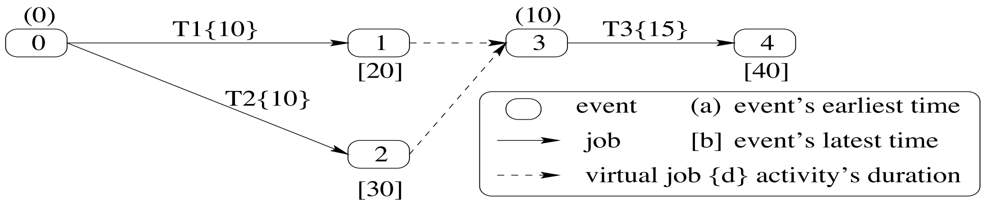

Figure 11: MQ level partial ordered schedule

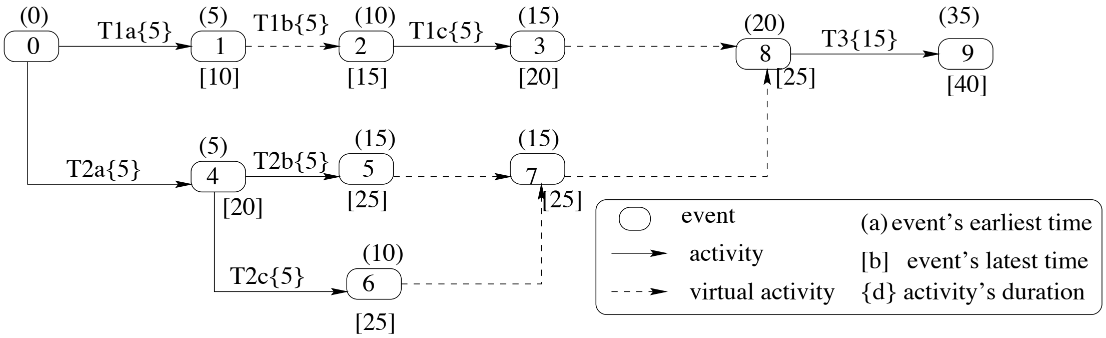

Figure 12: partial order schedule with one non-local subtask

tablished based on the TÆMS level schedule(TAG1_P5) and the MQ schedule(MQ_S1). Suppose in the TAG1_P5 schedule, the quality request of T1b is 10, and the abstraction of the schedule TAG1_P5 is (5, T1b, 5), it means there are some activities of duration 5 need to be done before T1b and some activities of duration 5 need to be done after T1b. The above information comes from the pre-analysis of the plan TAG1_P5. Combined with the temporal constraint of TAG1-P5 in the schedule MQ_S1 [0-20], the temporal scope of T1b is [5-15], it leaves 5 units time before T1b and 5 units time after T1b. These constraints are very preliminary, if there are other constraints added to other activities, the scope may need to be refined based on the TÆMS level rescheduling(See example in Section 4.7). The quality request is only an estimation because the agent does not know what quality the other agent may achieve for this task.

Agent A posts this task allocation proposal as:

(T1B, QUALITY-REQUEST:10 , TIME-SCOPE[5-15])

4.4 Partial Order Schedule

A partial order schedule is the basic reasoning tool for concurrent multiple related negotiations. Here I will present the formalization of the partial order schedule and use an example to explain how it works for the multi-linked negotiation. Figure 12 shows the partial ordered schedule with one nonlocal tasks T1b.

4.4.1 Graphic Structure of Partial Order Schedule

Definition PARTIAL ORDER SCHEDULE is represented as a graph $G = (E, A)$. $E = \{i\}, A = \{a :< i, j > |P(i, j) \bigwedge (i, j \in E)\}$. It is a directed non-circle graph.

Definition EVENT : A node i represents an event, it is the start or end point of one or more activities, it is the separation point of sequenced activities. The initial node only represents the start point of activity(activities), the terminal node only represents the end point of activity(activities). There is only one initial node and one terminal node in a partial order schedule graph, all other nodes represent both the start point and the end point of one or more activities.

Definition ACTIVITY : An edge a represents an activity. It is the basic element(task/method) of the schedule. An activity needs a certain amount of time and resources, it is represented as a solid line with an arrow in the graph. An activity edge $a :< i, j >$has an event i as the tail event(start point) and j as the head event(end point), i and j are the related events of this activity. $D(i, j)$ (also as $D(a)$) represents the duration of activity $a :< i, j >$.

Definition VIRTUAL ACTIVITY : An activity that does not consume any time or resource. It could be: (a) a nonlocal task/method; or (b) an edge that only expresses the relationship between events. It is represented as a dashed line with an arrow in the graph.

TIME OF EVENT

Definition EVENT'S EARLIEST TIME $E(j)$: When the event j is the tail event of one or more activities, $E(j)$ is the earliest start time of those activities; When the event j is the head event of one or more activities, $E(j)$ is the earliest finish time of those activities. The computation of $E(j)$ starts from the initial event $E(0)$, the earliest time of the head event j ($E(j)$) is the sum of the earliest time of the tail event i ($E(i)$) and the duration of the activity $< i, j >$ ($D(i, j)$). If there are more than one activity edge pointed to this head event j, first add each activity's duration to the earliest time of its tail event, then add these values together with counting the parallel execution between local activities and non-local activities($Sum - with - parallel - reduction$). Outside constraint refers to the earliest start time associated with those activities as extra information.

$E(0) = max(outside\_constraint(0), 0)$

$E(j) = max(outside\_constraint(j), Sum\_with\_parallel\_reduction\{E(i) + D(i, j)\})$ $(j = 1, 2, ..., n)$

Definition EVENT'S LATEST TIME $L(i)$: For a head event, it is the latest finish time of all activities ending at this event; for a tail event, it is the latest start time of all activities starting from this event. The computation of the latest time starts from the terminal event. The latest time of the tail event i is the latest time of the head event j subtract the duration of the activity $< i, j >(D(i, j))$. When there are more than one activities from this tail event i , the latest time of event i is the maximum of these events' latest times minus the sum of the durations of those activities (when those activities' execution are continue). The outside constraints(deadline constraints associated with activities) are also be taken into account.

$L(n) = outside\_constraint(n)$ n is the terminal event

$L(i) = min(outside\_constraint(i), max\{L(j)\} - Sum\_with\_parallel\_reduction\{D(i, j)\})$ $(i = n - 1, n - 2, ..., 1)$

TIME OF ACTIVITY

Definition ACTIVITY'S EARLIEST START TIME $ES(i, j) = E(i)$;

Definition ACTIVITY'S EARLIEST FINISH TIME $EF(i, j) = ES(i, j) + D(i, j)$;

Definition ACTIVITY'S LATEST FINISH TIME $LF(i, j) = L(j)$;

Definition ACTIVITY'S LATEST START TIME $LS(i, j) = LF(i, j) - D(i, j)$;

For example, in Figure 12, activity T_{2c} is actually the activity $(4, 6)$. $D(T_{2c}) = D(4, 6) = 5$.

$ES(T_{2c}) = ES(4, 6) = E(4) = 5$;

$EF(T_{2c}) = EF(4, 6) = ES(4, 6) + D(4, 6) = 5 + 5 = 10$;

$LF(T_{2c}) = LF(4, 6) = L(6) = 25$;

$LS(T_{2c}) = LS(4, 6) = LF(4, 6) - D(4, 6) = 25 - 5 = 20$;

FLEXIBILITY OF ACTIVITY

Definition TOTAL TIME DIFFERENCE OF ACTIVITY $TTD(i, j) = LF(i, j) - EF(i, j) = LS(i, j) - ES(i, j)$;

Definition SINGLE TIME DIFFERENCE OF ACTIVITY $STD(i, j) = ES(j, k) - EF(i, j)$. The duration that earliest start time of $< i, j >$can be postponed without affect the earliest start time of its following activity $< i, j >$;

Definition FLEXIBILITY OF ACTIVITY $F(a) = \frac{TTD(a)}{D(a)}$, $D(a)$ is the duration of activity a. The flexibility of the activity represents the freedom to move the activity around in this schedule.

Using the activity T_{2c} in Figure 12 as an example,

$TTD(T_{2c}) = LF(T_{2c}) - EF(T_{2c}) = LS(T_{2c}) - ES(T_{2c}) = 15$;

$STD(T_{2c}) = ES(6, 7) - EF(4, 6) = 15 - 10 = 5$;

Definition FLEXIBILITY OF SCHEDULE $F(S) = \sum_{a \in S} F(a) * \frac{D(a)}{D(S)}$, $D(S)$ is the total duration of the schedule S. The flexibility of a schedule measures the overall freedom of this schedule, it is the sum of the flexibility of each activity weighted by its duration of the duration of the schedule. The flexibility of the activity with a larger duration has a bigger influence on the flexibility of the schedule.

4.4.2 More definitions about a partial order schedule

Definition BREAKABLE EXECUTION ASSUMPTION: The assumption is that the agent can switch to another activity during the execution of one activity, and it can switch back at some point to continue the execution of this activity.

Definition FEASIBLE LINEAR SCHEDULE: A total ordered schedule of all activities with or without split activities, that fulfills following conditions:

- Each activity a takes n (n>=1) time periods $(t_i, i = 1, ...n$)for execution, $\sum_i t_i = D(a)$;
- All partial order relationships both from the MQ task level and from the TÆMS level are valid;
- All EST and DL constraints of MQ level tasks are valid;

Definition VALID PARTIAL SCHEDULE: Without additional constraint and with the breakable execution assumption, there exists at least one feasible linear schedule that can be produced from this partial schedule.

Definition NECESSARY CONDITIONED SCHEDULE If the activity a is executed out of the $[ES(a), LF(a)]$ period, there does not exist a feasible linear schedule for all activities.

Definition COMPLETE FREE COMMITMENT : Without additional constraint and with the breakable execution assumption, for an activity a with the range [EST, DL], no matter what time a is executed during this range, there exists at least one feasible sequential schedule that can be produced from this partial schedule, then the range [EST, DL] for activity a is a *complete-free-commitment* because activity a can be executed during any period in this range.

4.4.3 Related Algorithm

We need to build following algorithm to support the negotiation based on the partial ordered schedule.

1. generate a *necessary conditioned* partial order schedule from a set of linear schedules and a set of constraints;
2. verify if a partial order scheduler is valid given certain constraints;

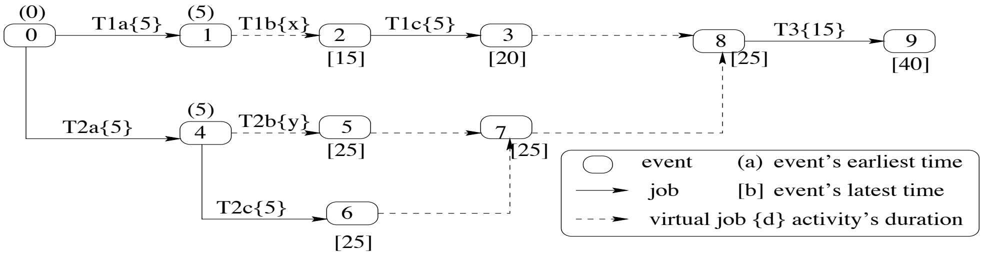

Figure 13: partial order schedule with two indirectly related issues

3. generate a linear schedule from a partial order schedule;
4. generate a new partial order schedule when adding a constraint - given a constraint, find its implication on other activities;
5. verify if a commitment is a *complete-free-commitment*;

4.5 Concurrent Multi-Linked Negotiation

When there are multiple related negotiations going on concurrently, the agent needs to analyze the relationships among these issues and find what is the influence of one issue on others. A partial order schedule is a reasoning tool that can help the agent deal with concurrent related negotiation.We use the following examples to explain how it works.

4.5.1 Indirectly Related Issues

In the example of section 4.3, there is a nonlocal subtask T1b in the plan TAG1_P5 (5, T1b, 5). In this plan, there are some activities with duration 5 that need to be finished before T1b can start, and there are other activities with duration 5 that only can be started after T1b is finished. This information comes from the analysis of the plan.

Suppose there is another nonlocal task T2b in the plan TAG2_P2 (5, [T2b, 5]) that needs to be contracted out. In this plan, there are some activities with duration 5 that need to be finished before T2b can start, and there are other activities with duration 5 that can be performed with T2b in parallel.

Figure13 shows the partial ordered schedule with two nonlocal tasks T1b and T2b. The largest possible range for T1b is [5, 15]; the largest possible range for T2b is [5, 25].

1. If these two ranges [5,15] for T1b and [5,25] for T2b are given to the contractee agents as the time constraints for the commitments, there is a local conflict. Because the task T1a needs to be finished before task T1b starts by time 5, and the task T2a needs to be finished before task T1b starts by time 5 also, both T1a and T2a need be to finished between [0,5]. Given each of them has a duration of 5, it is impossible to find a feasible local schedule.

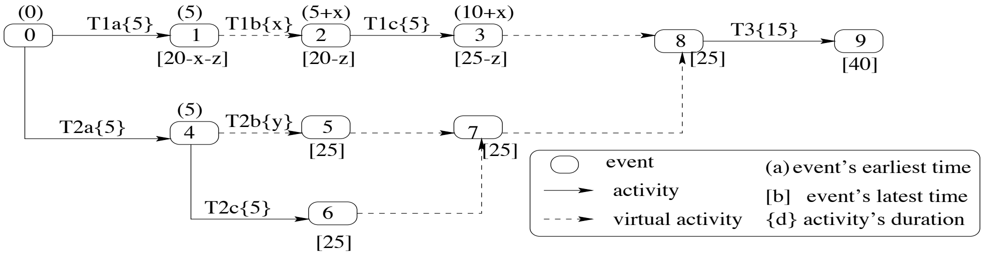

Figure 14: partial order schedule with two directly related issues

2. The two ranges [5,15] for T1b and [10,25] for T2b are consistent, that means, no matter what time those methods are performed during their ranges, there exists a feasible local schedule. Similarly, [10,15] for T1b and [5,25] for T2b are also consistent. The agent can do concurrent negotiations within these two consistent ranges, then the two negotiation processes can be independent.
3. If the agent gets the commitment [5,15] for T1b and [10,25] for T2b from the contractee agents, it may not be the best result. Although there exists a feasible schedule from the MQ level view, there is much less flexibility in the local schedule. Given these two commitments, the flexibility of tasks T1a, T2a, T1c and T3 are zero, which means the execution time for these tasks are totally fixed. If there is another constraint(i.e. a resource requirement for T2a) added, the agent may fail to find a feasible local schedule.

4.5.2 Directly Related Issues

Figure 14 shows an example with two *directly-linked* issues. Suppose another agent requests task T1 to be performed and the plan (T1a, T1b, T1c) is chosen to accomplish this task, where the subtask T1b needs to be contracted to another agent. The negotiation about T1b is directly related to the negotiation on T1. Here we only focus on the duration issue to study how these two linked negotiations should be processed.

Suppose there is no precondition for task T1, $ES(T_1) = 0$. Given the following task T3 with a deadline 40 and duration 15, the latest finish time of T1 is 25, $LF(T_1) = 25$. The largest possible range for T1 is [0, 25] and the largest possible range for T1b is [5, 20]. The negotiation on T1 is concerned about the finish time, the negotiation on T1b concerns about both the start time and the finish time.

1. Suppose x denotes the duration for task T1b, $25 - z$ denotes the deadline of task T1, x and z are two parameters that can be adjusted during the negotiation process. Following conditions need be satisfied: $10 + x \leq 25 - z; 5 \leq 20 - x - z \Rightarrow x + z \leq 15$.
2. Given $x = 10$, $z \leq 5$, [5,15] for T1b and [0, (20,25)] for T1 are two consistent ranges. The two negotiation issues can be performed within these two ranges concurrently.

Similarly, the impact of each commitment on local schedule and other commitments can be analyzed using the partial order scheduler.

3. Different values can be chosen for x and z, and other negotiation issues such as the non-local task T2b also can be taken into account at the same time. The agent could have an overview of all its on-going negotiation processes, hence it can find a better solution from global perspective.

4.5.3 General Ideas

From the above observations, we have following basic ideas for concurrent multi-linked negotiation:

1. Using the partial order schedule to find the largest possible range for each task in negotiation and the relationships among them;
2. Sorting these negotiation issues according to their flexibilities or their importance or the difficulties of negotiation processes and find the influence of the previous issue on the later issues hence find consistent ranges for those issues;
3. Concurrent negotiations can be performing within these consistent ranges without affecting each other;
4. The consistent range for one issue may be several discrete ranges(i.e. r1 and r2) rather than one bigger continue range, then the negotiation can be performed first in r1. If no commitment can be built in r1 or even if a commitment in r1 is found but proper commitments can not be found with constraint of r1, then r2 is being used as negotiation range. It is like a backtracking algorithm.
5. To speed up negotiation, the agent may take some risk of conflict at the MQ level negotiation, and try to solve them or avoid them at the TÆMS level negotiation. For example, the range r1 and r2 are almost consistent except a subrange (s_r1) and a subrange of r2 (s_r2) are conflict. If s_r1 is relatively small for r1 and so as s_r2 for r2, the probability of conflict is also very small, then the agent can take r1 and r2 as two consistent ranges and perform concurrent negotiations.
6. For the negotiation with a consistent range, the agent may not want to give the whole range at the first proposal (except if it wants to speed up the negotiation and get the negotiation resolved quickly). The agent will use part of the range as the first proposal and evaluate the proposal and counter proposal by finding out how it affects the flexibility of local schedule.

4.6 MQ level Negotiation

The negotiation on the MQ level includes multiple issues. The first issue is the MQ transferred when task NL is performed by one agent on the request of another agent. Another issue is the plan selected for performing task NL, including the start time, the completion time and the achieved quality of NL. Also, the agent needs to select an appropriate reward model that takes into account the possible further refinement of the rough commitment.

4.6.1 transferred MQ

There are three different models for the transferred MQ with the allocated task:

1. Fixed MQ. The type and the amount of the transferred MQ is fixed, it is determined by the organization relationship of the contractor agent and the contractee agent. There is no need to negotiate about the transferred MQ, but the contractor agent and the contractee agent may negotiate about the possible approach (certain quality and certain start time and finish time) of the task NL. This model implicitly represents authority relationship among agents.

2. Negotiated MQ. Agreed on certain approach to perform task NL, the contractor agent and the contractee agent negotiate about the type and amount of the transferred MQs depending on the contractor agent's current available MQ and the contractee agent's preference. The value of the transferred MQ would not exceed *marginal_ utility_ gain(NL)* (MUG) for the contractor and not less than the *marginal_ utility_ cost(NL* (MUC) for the contractee agent. For the contractee agent B, it schedules NL with its other MQ tasks based on the cost estimation, the *quality-request* and the *time-scope* of NL. Agent B evaluates the utility cost of performing NL as: *Marginal_ Utility_ Cost(NL) = Utility(the best schedule without NL) - Utility(the best schedule with NL)*

3. Dynamic MQ. The agents negotiate on both about the approach and the transferred MQ of NL. For every different approach to accomplish task NL, the *marginal_ utility_ gain(NL)* and the *marginal_ utility_ cost(NL)* are different, so the MQ value space for negotiation ([MUC, MUG]) is different too. Also for a certain approach, the agents may negotiate about the type and the amount of transferred MQ in the corresponding MQ value space as in model 2.

Figure 15 shows the two possible negotiation dimensions in MQ level. In model 1, the transferred MQ is fixed, the negotiation is only about the approach; in model 2, the approach is fixed, the agents are searching for an agreement point in the [MUC, MUG] scope by negotiating about the type and amount of transferred MQ; in model 3, the negotiation is performed on both dimensions.

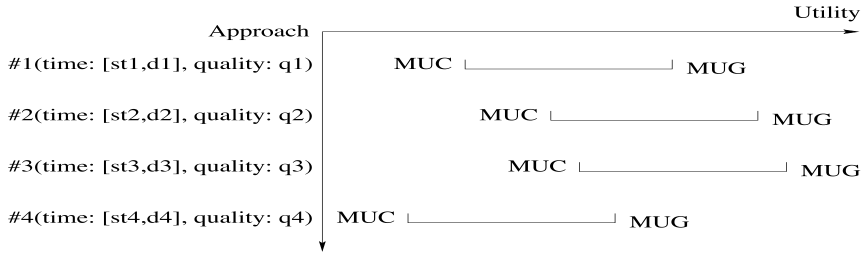

Figure 15: MQ level negotiation

4.6.2 Different Approaches

A certain approach specifies the lowest achieved quality, the earliest start time and the finishing time for the task. All these issues can be varied in the negotiation process to construct different approaches.

The contractor agent A builds the first proposal [NL, quality: q1, start time:s1, deadline: d1] based on the best MQ level schedule(MQ_S1) and the selected plan (TAG1_P5). This proposal is evaluated by the contractee agent B. If this proposal is rejected by agent B, it will return a counter proposal [NL, quality:q2, start time: s2, deadline: d2]. Agent A revises its local plan MQ_S1 based on this counter proposal and evaluates the utility of the new plan MQ_S1*.

There is a quality accumulative function that maps NL's quality to TA1's quality. This function is constructed based on the structure of TA1 and is available to the agent. The quality of TA1 is mapped to MQPS(TA1), that determines the utility of the new schedule MQ_S1*. If *MUC(counter_proposal) < Utility(MQ_S1*) - Utility(MQ_S2)* (MQ_S2 is the next best schedule, it may or may not need contract out NL) , the counter offer is acceptable; otherwise, it is rejected.

The agent A can reason about the influence of the start time and the completion time of NL in the counter proposal on its local plan using the partial order schedule. The influence includes: how it affects the temporal constraints of each tasks and how it affects the flexibilities of other activities and the local schedule.

Similarly, the agent B can evaluates the flexibility of the proposal(or the counter proposal) given the estimation of the execution time of the task. The flexibility can be calculated by comparing the [EST, DL] range to the duration of the task(d). As the range grows, the flexibility of the commitment becomes bigger. The following formula can be used to calculate the flexibility of a commitment($F(c)$):

$$F(c) = \frac{dl(c)-est(c)-d}{d}$$

In above example, suppose the estimated duration of NL is 5($d = 5$), the commitment c1 with range [5, 15] has freedom $F(c1) = 1.0$; the commitment c2 with range [10, 15] has freedom $F(c2) = 0$. When the flexibility of the commitment becomes bigger, it is easier for agent B to arrange its other local activities and achieve success on its other negotiation issues, but it is more difficult for agent A to arrange its other activities and get success on its other negotiations. So the value/cost of a commitment is also related to the flexibility of this commitment. If the agent A wants to keep more flexibility for itself and reduce the flexibility of the commitment, it needs to pay more to agent B. If agent A does not need too much local flexibility because it has a lot of certainty about its other local activities, it can give more flexibility to agent B which makes agent B's life much easier, then agent A can pay less for this easy commitment. Both agents can reason from their current states to decide if they need more flexibility or they need a cheap ("expensive" for agent B) commitment.

4.6.3 Rough Commitment and Rewards

Agents build rough commitments in the MQ level negotiation. A rough commitment specifies several issues for the request under negotiation(the contracted task in this context). The specification is a range rather than a point, which allows further refinement in this range. For example, a rough commitment c could specify the temporal constraint for the contracted task NL is $[t1, t2]$, if $F(c) > 0, t2 > t1 + d$(d is the duration of NL), it is possible to refine this commitment by restricting this range to $[t1 + x, t2 - y], (t2 - y - t1 - x \geq d)$, hence the flexibility of the commitment c is reduced. Because the flexibility is related to the value/cost of the commitment, the agents need to come to an agreement on how the latter refinement is related to the value of the transferred MQ. There are two possible models:

1. Pre-Paid model. The contract agent A pays $v1$ amount of MQ for the contractee agent B to perform task NL during any time period (not shorter than d) within $[t1, t2]$ as agent A requests . This agreement provides agent A great freedom on further refinement of the commitment, and agent B agrees to accommodate any request from agent A within the pre-defined range. No matter what request agent A will make or agent A does not make any further request, agent B will receive $v1$ amount of MQ as decided in the rough commitment.

2. Dynamic model. The contract agent A pays $v2$ amount of MQ for the contractee agent B to perform task NL within $[t1, t2]$. If agent A requests to restrict this range to $[t1 + x, t2 - y], (t2 - y - t1 - x \geq d)$ and if agent B could accept this request, agent A will pay $((x + y) * \beta + 1) * v2$ amount of MQ to agent B. Agent B would decide to accept this refinement request or not, according to its current problem-solving context. If agent B does not accept this request, it still is obliged to perform NL during $[t1, t2]$ and in turn is guaranteed to get $v2$ amount of MQ as the rough commitment defines.

These two models provide different degrees of freedom for the agents. The agents can choose a model according to the constraints and uncertainties of their local activities during the negotiation process.

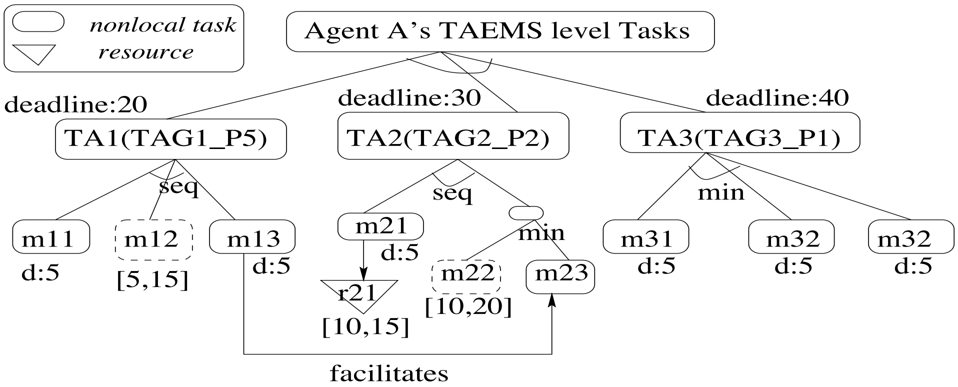

Figure 16: TÆMS level tasks

4.7 TÆMS level negotiation

Agent A reorders its TÆMS level tasks based on the plans chosen in the MQ level schedule. All methods not included in the MQ level schedule are eliminated from the task group and the tasks are associated with temporal constraints from the MQ level schedule.

Figure 16 shows agent A's current tasks and the required negotiation issues. Agent A currently has three tasks, task1, task2 and task3. All methods appears in this figure are those constructing the plan TAG1_P5, TAG2_P2 and TAG3_P1. Task1 has a deadline of 20, task2 has a deadline of 30 and task3 has a deadline 40. Task1 and task2 need to be finished before task3 starts. These constraints come from the MQ level scheduling. Also there are two commitments built on MQ level for the nonlocal methods m12$[5, 15]$ and m22$[10, 20]$. The agent tries to satisfy all these constraints when arranging its local activities. However, there may be other constraints that agent A needs to consider. These constraints come from the resource requirements and the relationships among those subtasks that belongs to different high-level tasks, they are not visible to the MQ level scheduler so are not reflected in the MQ level schedule. Two examples are shown in Figure 16 :

1. There is a facilitates relationship between m13 and m23. If agent A can complete m13 before it performs m23, the execution of m23 will be facilitated in terms of getting better quality, spending shorter duration or lower cost. So agent A needs to add this additional temporal sequence constraint $[m13 \rightarrow m23]$ into its partial order schedule if it wants to exploit this facilitates relationship(shown in Figure 17).

2. The execution of method m21 needs the resource r21. The resource r21 may be managed by a resource manager or may be shared with other agents. Agent A needs to find out what time r21 is available so it can arrange the execution time of method m21.

The reordering process considers all methods contained in the MQ level schedule. It takes into account the interrelationship among tasks, the resource request constraints and the

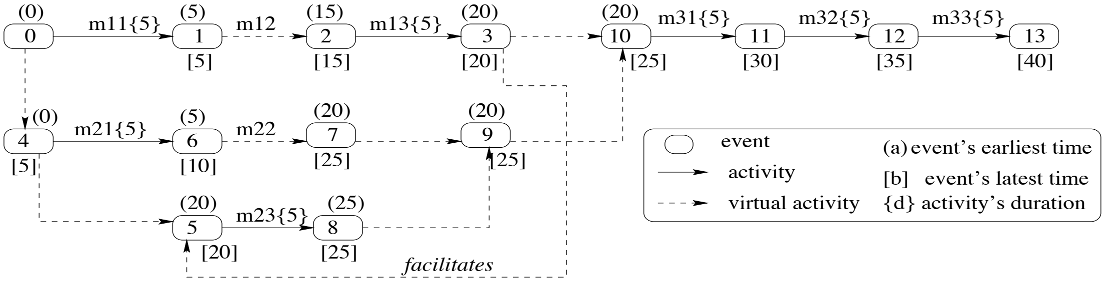

Figure 17: TÆMS level partial order schedule

rough commitments built at the MQ level negotiation. For example, resulting from the MQ level negotiation, agent B will perform task m12 for agent A between time 5 and 15, and agent C will perform task m22 for agent A between time 10 and 20. Given that the resource r21 is only available from time 10 to 15, agent A can not find a feasible local schedule. One solution is to negotiate with agent C to push the start time of m22 to 15 instead of 10 (suppose the duration of m22 for agent C is 5). If the commitment on m22 between agent A and C is the "*Pre-Paid model*", then agent C would accept this request. Otherwise, if the commitment is associated with the "*dynamic model*", agent C needs to reason about its local partial order schedule to find if it could grand this request. If yes, agent C will get extra MQ from agent A as they have agreed on the MQ level negotiation. If this refinement negotiation is successful, agent A could have a feasible local schedule:

```
m11[0-5] m12[5-15] m13[15-20]
m21[10-15] m22[15-20] m23[20-25]
m31[25-30] m32[30-35] m33[35-40]
```

4.8 MQ level rescheduling

If the refinement negotiation fails, agent A could not find a feasible local schedule given all local constraints, agent A has to redo the MQ level scheduling.

Before redoing the MQ level scheduling, the agent needs to make decision about what commitments should be preserved and what commitments should be decommitted. Some commitments are established and preserved, and the related MQ level tasks are associated with corresponding temporal constraints, other commitments are decommitted. This decision making process should take into account the importance of the commitments, the urgency of the commitments, the decommitment cost and the scarcity of the resource.

Suppose in the example of Figure 16, the agent decides to preserve the resource commitment for r21, because the resource r21 is scarce and all alternatives for TA2 include the method m21, then TAG2_P2 is associated with this new constraints: earliest_starting_time:10, and the MQ level scheduler will take this constraint into account for rescheduling.

- Mapping TÆMS level constraints to MQ level constraints

There is a technical problem of how to map the TÆMS level constraints to the constraints of MQ level tasks. For example, in Figure 16, the plan TAG2_P2 contains three methods m21, m22 and m23. The method m21 has to be performed from time 10 to time 15 given the resource constraint for r21. By analyzing the structure of TAG2_P2(m21 is the first method and other two methods are sequenced after it), it is known that the *earliest_start_time* of TAG2_P2 is 10 and its *deadline* is 30. Similarly, we can analyze the structures of other alternatives of task T2 and find the implication of m21's constraint on them. If the alternative plan does not contain method m21, then m21's constraint has no effect on this plan; otherwise , the influence depends on the structure of the plan. In this example, the structure is simple, so m2's constraint can be mapped exactly to TAG2_P2's constraint as *earliest_start_time = 10*. But when the situation is not so simple, if m21 and m22 are not sequenced but are unordered, then m22 can be done in anytime before time 10 or after 15, the earliest_start_time of TAG2_P2 is not fixed. (This is a question we still need to think.)

- The influence of the TÆMS level constraints on the MQ level scheduling

  After mapping the constraints of TÆMS level to MQ level task, we need think what it really means and how it affects the MQ level rescheduling. In the previous example, the m2's constraint is mapped to TAG2_P2's constraint as *earliest_start_time = 10*, but it does not mean TAG2_P2 only can be performed between time 10 and time 30. Another choice is to do TAG2_P2 after time 30. So TAG2_P2 can be split into two alternatives: TAG2_P2_1(*earliest_start_time* : 10, *deadline*: 30) and TAG2_P2_2(*earliest_start_time*: 30). When rescheduling, the MQ level scheduler uses these two alternatives instead of the plan TAG2_P2.

Another issue is about the influence of the decommitment cost. Suppose the agent has built the resource commitment for method m21 in the TÆMS level negotiation phase, there is a decommitment cost for it. This decommitment cost also should be mapped to the MQ level cost for those alternative plans that violate this commitment, i.e. TAG2_P2_2, it starts after time 30, the resource commitment for time [10,15] does not work for it, it needs break this commitment and builds another one. So when the MQ level scheduler reschedules, it needs take into account the decommitment cost.

4.9 Uncertainty

For simplicity of the description, uncertainty is omitted in section 4.4 when defining the partial order schedule. A single number instead of a statistical distribution is used to represent the duration of a task as the expected value, and the start times and deadline are also represented as a single number. Here we discuss the implication of the uncertainty and how to deal with it in the negotiation process.

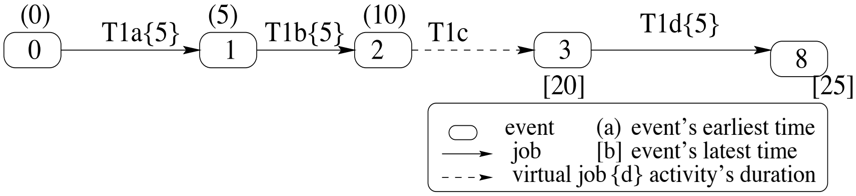

Figure 18: An example of one non-local task

4.9.1 Measure of Uncertainty

The uncertainty discussed in this section means the uncertainty in the estimation of the execution characteristics(i.e. duration, quality, and cost) of an activity. This type of uncertainty can be represented as a statistical distribution:

$V- \{p_1, v_1; p_2, v_2; ...; p_n, v_n\}$

it means variable V has chance of p_i to be the value of v_i $(i = 1, ...n)$.

Expected Value of variable V : $E(V) = \sum_i p_i v_i$

Measure of Uncertainty of variable V : $MU(V) = -\sum_i p_i * log(p_i) * \frac{|v_i - E(V)|}{E(V)}$

For example:

$X = \{0.3, 3; 0.4, 5; 0.3, 7\}$; $E(Y) = 5$; $MU(Y) = 0.125$

$Y = \{0.1, 4; 0.8, 5; 0.1, 6\}$; $E(Y) = 5$; $MU(Y) = 0.04$

X and Y has the same expected value, but X has bigger uncertainty than Y .

4.9.2 Handling of the Distribution Explosion Problem

When uncertainty is taken into account in the scheduling process, there is a distribution explosion problem. For example, in Figure 18, job T1c is a non-local task and jobs T1a, T1b and T1d are local tasks. From the above local schedule, the earliest start time of T1c is 10 and the deadline is 20. T1a, T1b and T1c all have the estimated duration 5. If we introduce uncertainty of the duration as a statistical distribution:

```
T1a: 5 {0.2 4, 0.6 5, 0.2 6}
T1b: 5 {0.1 3, 0.8 5, 0.1 7}
T1c: 5 {0.1 4, 0.8 5, 0.1 6}
```

The earliest time of event 2 will be :

```
(2) = T1a: 5 {0.2 4, 0.6 5, 0.2 6} +
      T1b: 5 {0.1 3, 0.8 5, 0.1 7}
    = {0.02 7, 0.06 8, 0.18 9, 0.48 10, 0.18 11,
       0.06 12, 0.02 13}
```

This is a distribution of seven possible values. The distribution could become more distributed when the reasoning processes is longer than two steps. The following are possible approaches for this problem:

- ignore the value of low possibility:

  `(2) = { 0.26 9, 0.48 10, 0.26 11}`

- cluster the similar values using the expected value:

  `(2) = {0.26 8.6, 0.48 10, 0.26 11.4}`

- preserve the marginal value(the minimum value and the maximum value in the distribution):

  `(2) = {0.02 7, 0.96 10, 0.02 13}`

If a single number is used in the negotiation process as the earliest start time of task T1c, it could be the expected value: 10 or the marginal value: 13.

The expected value reflects the estimation of the end time of the previous tasks, with more than 50% chance it works right. When the duration of the previous tasks exceeds the expected value, the agents need to coordinate their actions. This requires a refinement of the rough commitment and a real-time coordination. It is possible that the contractor agent to attach this distribution {0.74 10, 0.26 11.4} with the commitment to inform the contractee agent that with 26% chance it need to extend the start time to 11.4. If the contractee agent accept this commitment, it needs to prepare for this possible further refinement or coordination. On the other hand, the contractee agent could charge more service fee for this uncertainty.

The marginal value reflects the latest possible time to finish previous task. If this value is used to build the commitment, no future refinement or coordination is needed, but it makes the commitment unnecessarily over-constrained in most cases.

4.9.3 General Idea

The general idea to accommodate uncertainty in this negotiation framework is described as follows. In the low level reasoning process, the distribution explosion problem is handled as described in section 4.9.2. Uncertainty information is abstracted as the expected value, the marginal value and the measure of uncertainty. This abstraction information is used in the upper level reasoning process, the upper level process does not deal with the detailed distribution information. The measure of uncertainty and the marginal value attached with a commitment describes the possibility that a specified item in this commitment may need to be changed by the extent of the marginal value. If the contractee agent promise to accommodate this change when requested by the contractor agent, it could charge a higher price for this commitment, or an extra cost may be added when this change really happens(As the two models discussed in Section 4.6.3). Since the uncertainty is related to the cost of the commitment, agent needs to reason about how to appropriately balance between the risk of rescheduling and the cost of the negotiation process.

5 Anticipated Intellectual Contribution

1. Studying how the organizational relationships among agents(from cooperative to self-interesting) affect the negotiation strategies and the influence on the performance of the multi-agent system. By quantitatively reasoning about the relationship between agents, it is possible to study more general types of negotiation besides the self-interest negotiation and the cooperative negotiation in one framework, which enhances the ability of the multi-agent system to model more realistic problems and relationships.

2. Studying how to deal with the concurrent multi-linked negotiation issues explicitly, especially the negotiation chain problem(i.e. in the supply chain domain), which is a realistic problem in multi-agent system. Partial order schedule is a reasoning tool for agent to establish the concurrent negotiation scope for different issues; multi-leveled negotiation allows the refinement of the commitments to solve the possible conflicts among the related negotiation issues.

3. Development of a multi-leveled negotiation approach which allows negotiation to be performed on different abstraction levels with different emphasis. Negotiation on the upper level builds sketchy commitments which are the foundation for the global activity planning; negotiation on the lower level refines the commitments considering detailed constraints. Study how the multi-leveled negotiation improves the system performance by getting a better evaluation of a commitment and allowing uncertainty and more constraints adding in.

4. Development and analysis of a partial order schedule framework and the corresponding reasoning tools to support constructing a commitment, verifying a commitment and evaluating a commitment given complex constraints. As partial order scheduler is used in negotiation to speed up the convergence of the process and facilitate the concurrent negotiation for multi-linked negotiation issues.

5. Introducing flexibility into negotiation and quantitatively reason about the flexibility in the negotiation process. The agent can select different negotiation strategy according the flexibility of the negotiation issue. The agent also can decide where to maintain more flexibility given the prediction of the possible uncertainty of its local activities.

6. Negotiation is viewed as a multiple dimensional search process, agents are negotiating over multiple issues such as the temporal scope of the commitment, the cost of the commitment and the flexibility of the commitment rather than over one single issue. It is possible to find a commitment that is good for both agents given their different problem solving contexts and hence to maximize the joint utility of them.

6 Related Work

6.1 Contract Net Protocol and Bounded Rational Self-Interest(BRSI) Negotiation

The earliest work on negotiation can be traced back to Smith's work[53] on the contract net protocol(CNP). It is modeled on the contracting mechanism used by business to govern the exchange of goods and services. The contract net provides a solution for finding an appropriated agent to work on a given task based on the evaluation of the bids. This work established a basic negotiation protocol model involving the announcing-bidding-evaluation-awarding process. It allows a mutual selection by both managers and contractors, however, there was no formal model discussed in this work for making decisions about task announcing, bidding and awarding. The basic model of the negotiation process in our proposed work is also based on this work.

TRACONET[40] work extends the CNP work by introducing a formalization of the bidding and awarding decision process. This formalization uses the marginal cost calculations based on local agent criteria.The pricing mechanism generalizes the CNP to work for both cooperative and competitive agents. In addition, the CNP framework is extended to handle task interactions by clustering tasks into sets to be negotiated over as atomic bargaining item. Since an agent could generate multiple bids concerning different contracts pending concurrently in order to speed up the operation of the system, the choice of the bid price depends on which contracts the agent could actually receive. Safe and opportunistic pricing policies are compared: opportunism speeds up the negotiations, but safe policies guarantee monotonic decrease of the local cost. This work extends the basic CNP work by introducing decision making model and utility function calculation, however, the negotiation process does not allow counter proposal and the negotiation is focused only on one issue(the price).

Significant work[41] has been done to extend above work based on the use of the bounded rational self-interest agents(BRSI). The research work concerns BRSI agents deal intelligently with the uncertainty present in the negotiation process. The uncertainty comes from three facts: one is uncertainty of what events will come in the future; another is the agent is not aware of the results of concurrent ongoing negotiation events; the third is the approximate computation of utility. The first aspect of this work regards the negotiation protocol about the commitment formation process. Level commitment [41] [42] allows an agent to decommit by paying a decommitment penalty. Different leveled commitment protocols(Original contract [one task to move at a time]; Cluster contract[move two or more tasks between two agents]; Swap contract [two agents swap tasks]; Multi-Agent contract [at least three tasks to be transferred between at least three agents]) and their parameterizations are empirically compared to each other and to several full commitment protocols[1][2]. Concerning solution quality, the leveled commitment protocols are significantly better than the full commitment protocols of the same type, but the difference between the different leveled commitment protocols are minor. The second aspect of this work concerns the tradeoffs between negotiation risks and computation costs[3][43]. The implication of the deadline and timeouts on negotiation is also studied. In our work, the agents are assumed to be bounded rational but not necessarily to be self-interested. The techniques dealing with uncertainty in the above work are potentially helpful. For example, the idea of decommitment penalty fits to our work.

Hunsberger[15] and Sandholm[45] have studied combinational auction, in which there are multiple items for sale, participants who may place bids on arbitrary subsets of those items, and an auctioneer who must determine which awardable combination of bids maximizes revenue. It allows agents select a shared plan for the group though a distributed computation process. This work relates to our work where the agent has multiple issues to negotiate with multiple agents and tries to find a best combinational solution. The difference is there is no second round proposal in the auction, the bids are not changeable after they are sent out. Also the selection of awardable combination is a centralize process performed by one agent.

Collins etc.[7][8] introduce the concept of the flexibility to the contracting mechanism for the evaluation of bid. This is the only work we found related to the flexibility in negotiation. The bid evaluation process incorporates costs, task coverage, temporal flexibility, plan feasibility, and risk estimation. The contractee agent takes into account the flexibility of the commitment when it calculates the bid price. However, there is no explicitly reasoning about how to balance the flexibility and the feasibility like we propose in our research work.

Although multiple simultaneous negotiations are allowed in above work, the agent does not reason explicitly about the relationship among concurrent negotiation issues. A statistical model is used for the agent to predict future coming events when it evaluate current commitment. Also the relationship between two negotiation issues is simply the competition of agent's local resource(*indirectly-linked*), no negotiation chain that span over more than three agents is taken into account here.

6.2 Negotiation in Cooperative Distributed Problem Solving (CDPS)

In the cooperative distributed problem solving(CDPS) area (i.e. DENEGOT[32]), negotiation is viewed as a distributed search through potential compromises. To estimate the quality of potential solutions, the negotiation search space is structured into a lattice of sets of potential compromise solutions based on hard constraints. A solution in a higher set in the lattice, if it is achievable, will be preferable over a solution in a lower set. Agents first search under the hard constraint level representing the highest quality solution standard achievable in the current situation. By relaxing hard constraints, the set of compromises that qualify as a solution are enlarged. Agents search for a resolution under the relaxed hard constraint set when a solution can not be found under the current set of constraints. The limitation is the assumption that individual agents are only involved in one negotiation session at a time, a negotiation session must end before another agent can become involved.

Similarly Lander and Lesser[25] report about an approach called negotiation search: an algorithm that explicitly recognizes and exploits conflict to direct search activity across a set of agents. A set of states is defined which represent the states of the negotiation process with respect to the quality of the actual solution. The states are changed by applying negotiation operators to initiate, critique, extend, and relax solutions, and to terminate the search. Laasri et. al. describe a generic model, the recursive negotiation model(RNN) of multi-agent problem solving that details various situations that can potentially benefit from negotiation[27]. This model defines where and how negotiation can be applied during problem solving based on structuring problem-solving into four stages: problem formulation, focus-of-attention, allocation of goals or tasks to agents, and achievement of goals or tasks. For the Belief-Desire-Intention(BDI) agent framework, a collaborative negotiation process[5][6] is used to resolve the conflicts in the shared plan of actions and the shared beliefs. It is a recursive *Propose-Evaluate-Modify* cycle of actions though that agents exchange evidence and justification to adopt their beliefs. In these works, negotiation is used to solve conflict in a cooperative problem-solving environment, agents need to work synchronously on their problem-solving negotiation process. No task or resource transferred among agents by negotiation, and agents don't need to explicitly reason about the utility of alternative proposals/counter-proposals.

A multistage negotiation paradigm is used for solving distributed constraint satisfaction problem[9]. Agents first make tentative commitments to local alternatives that would serve to partially satisfy the goals for which they are responsible. In making a tentative commitment, an agent removes the affected resources from the pool of available resources. After making a tentative commitment, an agent requests that other agents confirm this commitment by making their own local commitments in ways that are compatible. If some agents can not do so, an iterative exchange that results in the construction of induced exclusion sets and goal exclusion sets in the agents. The process of attempting satisfaction of multiple global goals occurs simultaneously in the network. It is sometimes necessary to retry alternatives that had previously failed when tentative commitments have been retracted. This idea is extended and applied to the distributed meeting scheduling problem.

In the distributed meeting scheduling process[47][48][50],a host agent announces the meeting time, it may be one option or several options, other attendant agents look at their scheduler to find a "yes/no" answer or provide other alternatives. The commitment strategies could be committed(tentatively blocks the time it proposed) or non-committed(not block time until an agreement is reached). Multiple meeting scheduling processes could be going on concurrently, the negotiation process on one meeting would affect other meetings, hence the commitment strategy is an important factor which affects the system's performance. Adaptive choice of commitment is recommended according to the combination of environmental factors. The choice of search biases in scheduling meetings also affects the conflicts frequency between new processes and already finished meeting scheduling processes. The problem domain studied here is a relatively narrow domain and the relationship among multiple negotiation issues is the competition for local resource(time). The policy to reduce conflicts is to choose appropriate strategy from the statistical view, no explicitly reasoning about the particular related issues when building proposal and evaluating proposal.

Progressive negotiation[29][30] is proposed to deal with the negotiation involves multiple homogeneous agents. The negotiation among a group of agents can be divided into a number of sub-negotiation which proceed incrementally. The agents are cooperative and can always negotiate to increase their mutual benefits. This approach is not suitable for our problem because we have a more general problem domain and more complex agent systems.

Above work focus on the research of the negotiation process model. All agents are assumed cooperative and working on a common goal. Negotiation is mainly used for conflict resolution.

6.3 Game Theory in Negotiation

Game theory provides a formal framework for negotiation. However, it frequently makes simplifying assumptions to facilitate the mathematical analysis. This approach includes building a formal model of negotiation, developing appropriate negotiation protocols and strategies and analyzing their equilibrium. Traditionally, game theory can be divided into two branches:cooperative and non-cooperative game theory. Cooperative game theory abstracts away from specific rules of a game and is mainly concerned with finding a solution given a set of possible outcomes. A topic like coalition formation is typically analyzed using cooperative game theory. In such games, a surplus is created when two or more players cooperate and form a coalition. Cooperative game theory can then determine how the surplus is to be divided given a coalition and a set of assumption. Non-cooperative game theory, on the other hand, is concerned with specific games with a well defined set of rules and game strategies. All strategies and rules are known beforehand by the players. The notion of an equilibrium strategies to determine "rational" outcomes of a game.

Zlotkin and Rosenschein import game theoretic techniques into the MultiAgent negotiation ([58] [59] [60] [61] [62]). They studied the task-oriented domain, the state-oriented domain and the worth-orient domain and explored the relationship between certain domain characteristics and the negotiation mechanism.

Based on this game theoretic approach, Kraus, Schwartz and Wilkenfeld developed a strategic model of negotiation that takes the passage of time during negotiation process itself into consideration ([21] [52]). Changes in the agent's preferences over time will change their strategies in the negotiation and , as a result, the agreements they are willing to reach. This model is applied to the data allocation problem in MultiAgent environment.

However, several assumptions limits the practical applicability of game-theoretic results. Also these assumptions distinguish this work from our work. Common assumptions are for instance: (1) complete knowledge of the circumstances in which the game is played. The rule of the game and the preferences and beliefs of the players are "common knowledge". (2) full rationality of the players. *Isolated Negotiation* is also assumed in this work, each negotiation stands alone, no relationship between any two negotiation issues.

6.4 Behavioral Science in Negotiation

Behavioral and social sciences study human cooperation and coordination and develop frameworks and models of organizations and communities. Their theory and knowledge are beneficial for the research of negotiation. Sycara presents the PERSUADER system [54] that solves conflicts though direct negotiation among the interacting agents or though a third party, the mediator. It is based on the case-based reasoning and multi-attributed utility theory. Argumentation [37] is introduced as negotiation mechanism for the agent with BDI (belief-desire-intentions) architecture. It is an iterative process that emerges from exchanges among agents to persuade each other and bring about a change in intentions, so it can be used for achieving cooperation and agreement. The above work introduces the idea that the agents can change their criteria functions though information exchange during the negotiation process, when they find another criteria function that is more suitable for current situation. This idea is useful to our proposed work.

Faratin, Sierra, and Jennings [12] present a formal model of an autonomous agent's decision function as it related to the process of service-oriented negotiation. The negotiation is about many issues between two parties. The model defines a number of tactics (time-dependent, resource-dependent, and behavior dependent) which the agent can employ during negotiation to generate offers and counter offers, evaluate proposals and offer counter proposals. Also this model indicates how an agent can change these tactics over time given strategic behavior. This model of negotiation is applied in the business process management system ADEPT [16] [17]. The idea that the agent dynamically chooses negotiation tactics is similar to our approach that the agent chooses negotiation strategy in real-time based on the relationship with another agent and the issue under negotiation.

6.5 Learning in Negotiation

Several learning technologies are introduced into negotiation to improve the negotiation performance. When agents have incomplete information about each another, it becomes important to learn about the other agent by observing their behavior during negotiation. A simple learning method based on Bayesian classifier allows an agent to make predictions about other agents' preference by building statistical models of other agents; preference functions from its past interactions with them [4]. The learning mechanism helps to reduce the amount of communication needed, and thus improve the overall efficiency of the negotiation process. Similarly, Bayesian beliefs are used to learn about other agent's reservation price[57], these beliefs are updated depending on the opponent's moves. However, once both agents use beliefs to determine their strategies, they also need beliefs about their opponent's beliefs, and so on. This is known as the problem of "outguessing regress". In practical applications, this problem is circumvented by assuming bounded rationality.

The generic program approach is the basic technique used to make the negotiation systems adaptive[31] [36]. Oliver [36] demonstrates a system of adaptive agents that can learn effective negotiation strategies. Computer simulations of "alternative-offers" negotiations are presented. Binary coded strings represent the agents' strategies. Two parameters are encoded for each negotiation round: a threshold which determines whether an offer should be accepted or not and a counter offer in case the opponent's offer is rejected. These elementary strategies were then updated in successive generations by a genetic algorithm.

Above learning technique also can be applied to our work for agents to learning the relationships between each other and hence be able to select appropriate strategy.

References

[1] Andersson, M. and Sandholm, T. 1998. *Leveled Commitment Contracting among Myopic Individually Rational Agents.* Proceedings of the Third International Conference on Multiagent Systems (ICMAS), pp. 26-33, Paris, France.

[2] Andersson, M. and Sandholm, T. 1998. *Leveled Commitment Contracts with Myopic and Strategic Agents.* Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI), pp. 38-45, Madison, WI, July.

[3] Andersson, M. and Sandholm, T. 1999. *Time-Quality Tradeoffs in Reallocative Negotiation with Combinatorial Contract Types.* National Conference on Artificial Intelligence (AAAI), pp. 3-10, Orlando, FL.

[4] H. H. Bui and S. Venkatesh and D. Kieronska *Learning Other Agents' Preferences in Multiagent Negotiation using the Bayesian classifier* , International Journal of Cooperative Information Systems, Vol. 8 Nos. 4, pp275-293, 1999.

[5] Jennifer Chu-Caroll and Sandra Carberry *Response Generation in Collaborative Negotiation* Proceedings of the Thirty-Third Meeting of the Association for Computational Linguistics, pp. 136-143, Morgan Kaufmann, 1995.

[6] J. Chu-Carroll and S. Carberry *Conflict Detection and Resolution in Collaborative Planning* Intelligent Agents II (LNAI 1037), pp. 111-126, Springer-Verlag: Heidelberg, Germany, 1996.

[7] John Collins, Scott Jamison, Bamshad Mobasher, Maria Gin *A Market Architecture for Multi-Agent Contracting* Proceedings of the Second International Conference on Autonomous Agents (Agents'98), pp. 285-292, ACM Press, 1998.

[8] John Collins and Maksim Tsvetovas and Rashmi Sundareswara and Joshua van Tonder and Maria Gini and Bamshad Mobasher *Evaluating risk: flexibility and feasibility in multi-agent contracting* Proceedings of the Third International Conference on Autonomous Agents (Agents'99), pp. 350-351, ACM Press, 1999.C

[9] 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.

[10] Decker, K., Lesser, V. R. *Quantitative Modeling of Complex Environments.* In International Journal of Intelligent Systems in Accounting, Finance and Management. Special Issue on Mathematical and Computational Models and Characteristics of Agent Behavior., Volume 2, pp. 215-234, 1993.

[11] Edmund H. Durfee and Victor R. Lesser *Negotiation Through Partial Global Planning* Proceedings of the 1988 Distributed AI Workshop, May 1988.

[12] Peyman faratin, Carles Sierra, Nick R. Jennings. *Negotiation Decision Functions for Autonomous Agents* Int. Journal of Robotics and Autonomous Systems, Vol. 24, Number 3-4, pp. 159-182, 1998.

[13] E. H. Gerding and D. D. B. van Bragt and J. A. La Poutrĺę *Scientific Approaches and Techniques for Negotiation A Game Theoretic and Artificial Intelligence Perspective*, technical report, January 2000.

[14] R. Guttman and P. Maes. *Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce.*Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98). Paris, France, July 1998.

[15] Luke Hunsberger and Barbara J. Grosz. *A Combinatorial Auction for Collaborative Planning.* In Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000). To appear.

[16] N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien and B. Odgers. *Autonomous Agents for Business Process Management.* Int. Journal of Applied Artificial Intelligence, 2000.

[17] N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, B. Odgers and J. L. Alty. *Implementing a Business Process Management System using ADEPT: A Real-World Case Study.* Int. Journal of Applied Artificial Intelligence, 2000.

[18] G. E. Kersten and T. Szapiro *Generalized Approach to Modelling Negotiations* European Journal of Operational Research, Vol. 26, pp. 142-149, 1986.

[19] Sarit Kraus and Eithan Ephrati and Daniel Lehmann *Negotiation in a non-cooperative environment* Journal of Experimental and Theoretical Artificial Intelligence, 3(4), pp. 255-282, 1991.

[20] .S. Kraus and J. Wilkenfeld. *A Strategic Negotiations Model with Applications to an International Crisis.* IEEE Transactions on Systems, Man, and Cybernetics, 23(1):313-323 1993.

[21] Sarit Kraus, Jonathan Wilkenfeld, Gilad Zlotkin *Multiagent Negotiation Under Time Constraints* Artificial Intelligence 75(2):297-345 June 1995

[22] S. Kraus and D. Lehmann. *Designing and Building a Negotiating Automated Agent,* Computational Intelligence, 11(1):132-171, 1995

[23] S. Kraus, K. Sycara and A. Evenchik. *Reaching agreements through argumentation: a logical model and implementation.* Artificial Intelligence journal, 104(1-2):1-69, 1998.

[24] Kazuhiro Kuwabara and Victor R. Lesser *Extended Protocol for Multi-stage Negotiation* Proceedings of the Ninth Workshop on Distributed AI, September 1989.

[25] 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.

[26] Brigitte Lassri and Hassan Lassri and Victor R. Lesser *Negotiation and Its Role in Cooperative Distributed Problem Solving* Proceedings of the Tenth International Workshop on Distributed AI, October 1990.

[27] Laasri, B., Laasri, H., Lander, S. and Lesser, V. *A Generic Model for Intelligent Negotiating Agents.* In International Journal on Intelligent Cooperative Information Systems, Volume 1, Number 2, pp. 291-317, 1992.

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

[29] Lyndon C. Lee *Progressive multi-agent negotiation* Proceedings of the First International Conference on Multi–Agent Systems, MIT Press, 1995.

[30] Lyndon Lee, *A Model of Progressive MA Negotiation* Proc. 3rd Int. Conf. on Multi-Agent Systems (ICMAS-98), pp. 447-448, 1998.

[31] N. Matos and C. Sierra and N. R. Jennings *Determining successful negotiation strategies: an evolutionary approach* Proc. 3rd Int. Conf. on Multi-Agent Systems (ICMAS-98), pp. 182-189, 1998.

[32] Moehlman, T., Lesser, V., and Buteau, B. *Decentralized Negotiation: An Approach to the Distributed Planning Problem.* In Group Decision and Negotiation, Volume 1, Number 2, pp. 161-192, 1992, K. Sycara (ed.), Norwell, MA: Kluwer Academic Publishers.

[33] H. J. MuĺŹller *Negotiation Principles* Foundations of Distributed Artificial Intelligence, John Wiley and Sons, 1996.

[34] Neiman, D. and Lesser, V. *A Cooperative Repair Method for a Distributed Scheduling System.* In AIPS-96, Edinburgh, Scotland, May, 1996.

[35] D. Neiman, D. Hildum, V. R. Lesser, and T. W. Sandholm, *Exploiting meta-level information in a distributed scheduling syste*, In Proc. Twelfth National Conference on Artificial Intelligence , 1994.

[36] 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.

[37] S. Parsons, C. Sierra and N. R. Jennings. *Agents that reason and negotiate by arguing.* Journal of Logic and Computation 8 (3) 261-292, 1998.

[38] Zhun Qiu and Milind Tambe and Hyuckchul Jung *Towards flexible negotiation in teamwork* Proceedings of the Third International Conference on Autonomous Agents (Agents'99), pp. 400-401, ACM Press, 1999.

[39] Sandholm, T. 1993. *An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations.* Eleventh National Conference on Artificial Intelligence (AAAI-93), Washington DC, pp. 256-262.

[40] 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.

[41] Sandholm, T. and Lesser, V. 1995. *On Automated Contracting in Multi-enterprise Manufacturing.* Proceedings of the "Improving Manufacturing Performance in a Distributed Enterprise: Advanced Systems and Tools" Conference, Edinburgh, Scotland, pp. 33-42, July 13-14.

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

[43] Sandholm, T. and Vulkan, N. 1999. *Bargaining with Deadlines.* National Conference on Artificial Intelligence (AAAI), pp. 44-51, Orlando, FL.

[44] Tuomas W. Sandholm, D*istributed Rational Decision Making*, Chapter 5, Multiagent system, editor Weiss. 1999

[45] Sandholm, T. and Suri, S. 2000. *Improved Algorithms for Optimal Winner Determination in Combinatorial Auctions and Generalizations.* National Conference on Artificial Intelligence (AAAI), Austin, TX, July 31-August 2.

[46] N. Sadeh-Koniecpol, D. Hildum, D. Kjenstad, and A. Tseng, *MASCOT: An Agent-Based Architecture for Coordinated Mixed-Initiative Supply Chain Planning and Scheduling.* Workshop notes, Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, Third International Conference on Autonomous Agents (Agents '99), May 1999.

[47] Sandip Sen and Edmund H. Durfee *The role of commitment in cooperative negotiation* International Journal of Intelligent and Cooperative Information Systems, 3(1), pp. 67-81, 1994.

[48] Sandip Sen and Edmund H. Durfee, *A Contracting Model for Flexible Distributed Scheduling,* Annals of Operations Research, volume 65, pages 195-222, 1996.

[49] 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.

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

[51] Sandip Sen and Anish Biswas, *More than envy-free* in the Working Papers of the AAAI-99 Workshop on Negotiation: Settling Conflicts and Identifying Opportunities, pages 44-49. (Workshop Notes available as AAAI Technical Report WS-99-12).

[52] R. Azulay-Schwartz and S. Kraus. *Negotiation On Data Allocation in Multi-Agent Environments.* Autonomous Agents and Multi-Agent Systems journal (to appear).

[53] Reid G. Smith: *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver.* IEEE Transactions on Computers 29(12): 1104-1113 (1980)

[54] Katia Sycara *Multi-Agent Compromise via Negotiation* Distributed Artificial Intelligence, Volume II, Pittman Publishing Ltd and Morgan Kaufmann, 1989.

[55] Thomas Wagner, Alan Garvey, and Victor Lesser. *Criteria-Directed Heuristic Task Scheduling.* Intl. Journal of Approximate Reasoning, Special Issue on Scheduling, 19(1-2):91–118, 1998. A version also available as UMASS CS TR-97-59.

[56] Wagner, Thomas and Lesser, Victor. *Relating Quantified Motivations for Organizationally Situated Agents.* In Intelligent Agents VI: Agent Theories, Architectures, and Languages, To appear 2000, also available as UMASS CS Tech Report 1999-21, April, 1999, Springer

[57] Dajun Zeng and Katia Sycara *Benefits of Learning in Negotiation* Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97), pp. 36-42, AAAI Press, July 27-31 1997.

[58] Gilad Zlotkin and Jeffrey S. Rosenschein *Negotiation and Task Sharing Among Autonomous Agents in Cooperative Domains* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pp. 912-917, August 1989.

[59] Gilad Zlotkin Jeffrey S. Rosenschein *A Domain Theory for Task Oriented Negotiation* (1993) In Proc. 13th International Joint Conference on Artificial Intelligence,14, 416-422, 1993

[60] Gilad Zlotkin and Jeffrey S. Rosenschein *Mechanism Design for Automated Negotiation, and its Application to Task Oriented Domains* Artificial Intelligence, 86(2), pp. 195-244, 1996.

[61] Gilad Zlotkin and Jeffrey S. Rosenschein *Compromise in Negotiation: Exploiting Worth Functions Over States* Artificial Intelligence, Vol. 84, Number 1-2, pp. 151-176, 1996.

[62] Gilad Zlotkin Jeffrey S. Rosenschein *Mechanisms for Automated Negotiation in State Oriented Domains* , Journal of Artificial Intelligence Research 5 (1996) 163-238.