A Proposed Approach to Sophisticated Negotiation *

Xiaoqin Zhang, Victor Lesser
Department of Computer Science
University of Massachusetts at Amherst
xqzhang@cs.umass.edu

Tom Wagner
Department of Computer Science
University of Maine
Orono, Maine 04469

Abstract

This paper proposes an approach to attack the multi-linked negotiation in a complex organizational context. The motivational qualities (MQ) framework developed earlier, provides the agent with the capability to reason about different objective goals hence the agent can evaluate a negotiation issue via its organizational objectives. The partial order schedule is exploited as a reasoning tool for the agent to handle the concurrent multiple linked negotiation issues and evaluate the flexibility and the feasibility in the negotiation. 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. Also, we bring multiple issues into the negotiation process such as the temporal scope of the commitment, the cost of the commitment and the flexibility of the commitment. The agents are negotiating over multiple issues rather than over a single issue.

Keywords: Multi-Linked, Multi-Leveled, Integrative Negotiation;

1 Introduction

Negotiation, an interactive communication among participants to facilitate a distributed search process, it is 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.

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 (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 producer agent and the supplier agent has a direct influence on the negotiation between the producer 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 task 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 [4] allows an 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 retract from its concurrent commitment by paying a decommit penalty to get an increase on its local utility. In a distributed meeting scheduling problem [5] [6], 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. However, in both of these works, 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.

Another question we want to address in this research is negotiation in a complex organizational context. Until now all related work has concerned either self-interested negotiation or cooperative negotiation [2]. 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 purely 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 Systems 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 purely self-interested to totally cooperative. The agents can choose to form a virtual organization [3] 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 focus our research

*This material is based upon work supported by the National Science Foundation under Grant No. IIS-9812755, DMI-0122173 and the Air Force Research Laboratory/IFTD and the Defense Advanced Research Projects Agency under Contract F30602-99-2-0525. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory/IFTD, National Science Foundation, or the U.S. Government.

work on following three categories of negotiation. In the *decision* category, we will use the motivational qualities (MQ) [8] 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 allows agents to take different positions anywhere along the spectrum from the self-interested position to the cooperative position. 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 [8] and the TÆMS framework [1] that are used as supporting architectures in this research. However, the major ideas are not restricted to these two frameworks, we feel they can also be applied to other suitable architectures.

2.1 MQ framework

In the MQ framework, the execution of a task contributes, in a quantitative 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 preferences and organizational goals or directives.

MQ Tasks are abstractions of a partial order 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 [1] (See Figure 2) 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 the 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 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 versus the importance it attaches to helping other agents increase their utilities.

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 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.

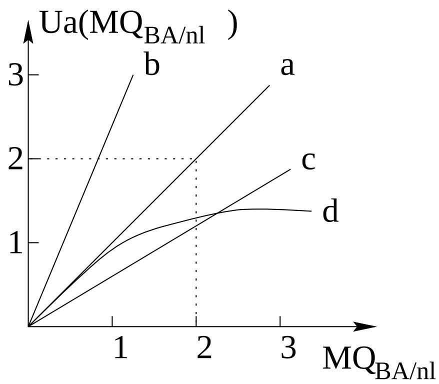

Figure 1: different mapping functions of $MQ_{BA/nl}$

The motivational qualities (MQ) [8] 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 with agent B on the concern 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't be transferred to any other kind of MQ, and it can't be transferred to any other agent; in other word, $MQ_{BA/nl}$ is useless for agent A in the market, it is simply to measure the relationship between agents A and B. Actually, how $MQ_{BA/nl}$ is mapped into agent A's utility depends on how cooperative agent A is with agent B. Suppose there is a certain amount, $MQ_{BA/nl}$, representing the utility agent B gained by having agent A perform task nl, transferred to agent A. Figure 1 shows four different functions for mapping $MQ_{BA/nl}$ to agent A's utility.

Function a, b and c are linear 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 fully cooperative with agent B;

If $k > 1$(b), $U_a(MQ_{BA/nl}) > MQ_{BA/nl} = U_b(nl)$, then agent A is accommodating to agent B, or altruistic with agent B;

If $k < 1$(c), $U_a(MQ_{BA/nl}) < MQ_{BA/nl} = U_b(nl)$, then agent A is partially cooperative (in contrast to fully cooperative) with agent B ;

If $k = 0$, $U_a(MQ_{BA/nl}) = 0$, 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 also be a nonlinear function (d) that describes a more complicated attitude of agent A to agent B, for example, agent A being fully cooperative with agent B for some period and then becoming self-interested. 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. An agent can adjust the utility mapping function to reflect its relationship with another agent, which could be it's administrator, colleague, friend, client or competitor. By adjusting some parameters in the mapping function, more subtle relationships could be managed. The agent could differentiate a friendly colleague from an unfriendly colleague, also it could draw distinctions between a best friend and an ordinary friend.

3.2 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 two available plans, P1 and P2, each one having different quality, duration and cost characteristics. The agent gets 5 units extra local utility by adopting plan P2 other than plan P1. However, plan P2 requests assigning a subtask Mc to another agent. From the high level view, if agent A can find another agent to perform the subtask Mc in time and with transferred utility less than 5, the plan P2 is the best choice. If such a commitment is not available, agent A needs to choose plan P1 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 activity 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 that 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 another agent), should be negotiated first at the upper level, and rough commitments should be constructed for them. However, those issues that have less influence on local plan selection and involve reasoning about the detailed structure of the low level activities can't 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 belong 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

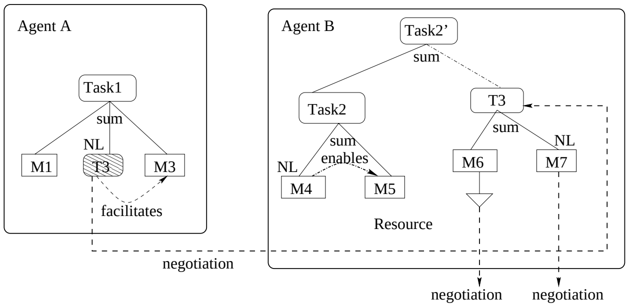

Figure 2: Agent A has a nonlocal task T3 contracted to agent B while agent B needs to subcontract M7 (a subtask of T3) to another agent and request resource for M6 (another subtask for T3) through negotiation.

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 a plan about the task. Expected values 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 situations detailed reasoning about uncertainty becomes important to making a commitment. The lower level has detailed information about the uncertainty, and as more context knowledge is available along with the process, so 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 take 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 its local plan and commitments, reduce failure possibilities, avoid conflicts and achieve higher utilities.

3.3 Multi-Linked Negotiation

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

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 and 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 2 describes a *directly-linked* relationship, Figure 3 describes a

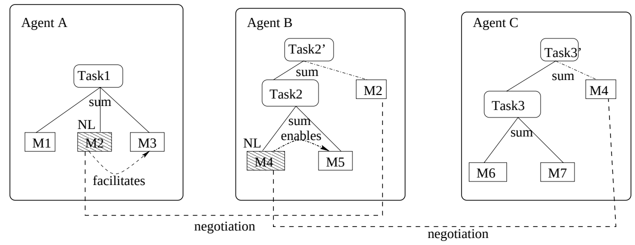

Figure 3: Agent A has a nonlocal task M2 contracted to agent B while agent B has a nonlocal task M4 contracted to agent C.

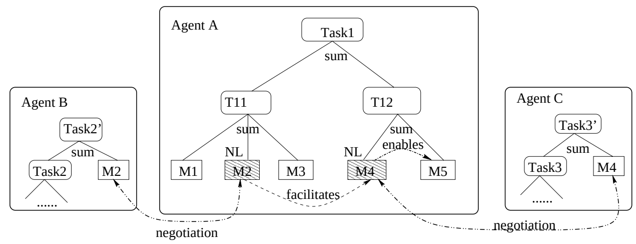

Figure 4: Agent A has two nonlocal tasks: M2 contracted to agent B and M4 contracted to agent C.

indirectly-linked relationship, and Figure 4 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 4, 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 negotiates 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't 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.

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.

4 Approach

In this section, examples will be developed to explain in detail how the multi-leveled, multi-linked integrative negotiation

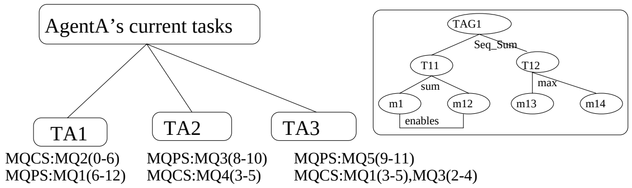

Figure 5: An example of MQ tasks

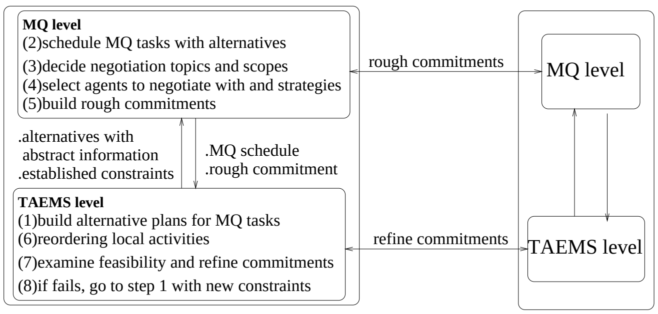

Figure 6: two level negotiation framework

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 [8] 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 [1] 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, which is a set of potential MQ tasks, each 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 5 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 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 6 , 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 to 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 an external resources needed by one of its tasks). Negotiation on the 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 the 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 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't 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 the following sections.

4.2 DTC scheduler builds alternatives

The Design-To-Criteria (DTC) scheduler [7] is a domain-independent scheduler that aims to find a feasible schedule that matches the agent's particular criteria request. In this research, 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.

The DTC scheduler works on TAG1 according to the 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 7. Each plan has different q, c, d characteristics, corresponding 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.

This abstraction process can be done off-line, and these alternative plans can be stored in the agent's database. Not all

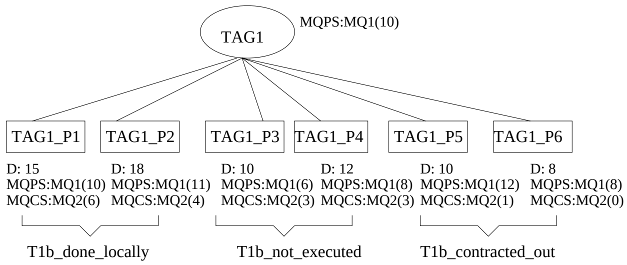

Figure 7: TAG1's alternatives

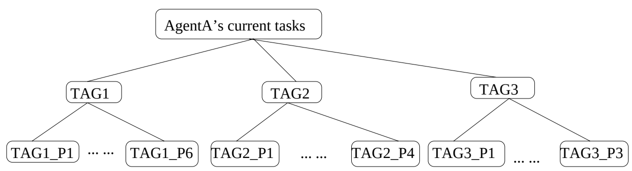

Figure 8: MQ level tasks

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

The MQ level scheduler does scheduling for these alternatives of TA1, TA2 and TA3 to find the best schedule MQ_S1 that provides the agent the most utility increase from its current state (Figure 8). If the plan TAG1_P5 or TAG1_P6 (T1b is contracted out) appears in the scheduler MQ_S1, agent A needs to 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 9. 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 another 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 established based on the TÆMS level schedule (TAG1_P5) and the MQ schedule (MQ_S1). Suppose

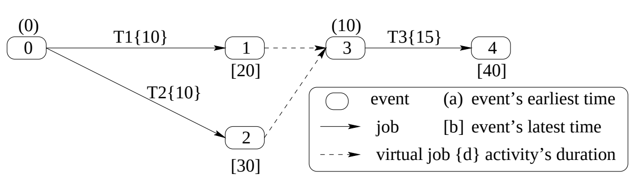

Figure 9: MQ level partial order schedule

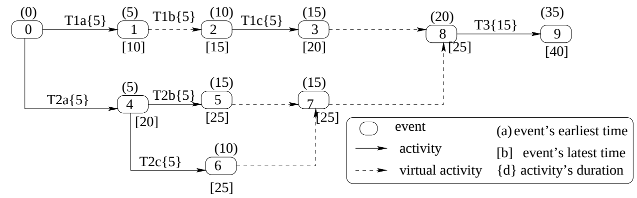

Figure 10: partial order schedule with one non-local subtask

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 we will present the formalization of the partial order schedule and use an example to explain how it works for the multi-linked negotiation. Figure 10 shows the partial order schedule with one nonlocal task T1b.

Definition *Partial Order Schedule is represented as a graph* $G = (E, A)$. *E represents nodes, each node denotes an Event (see below for definition); A represents edges, each edge denotes an Activity (see below for definition).* $E = \{i\}, A = \{a :< i, j > |(i, j \in E)\}$. *It is a directed acyclic graph (DAG).*

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 10, 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 10 as an example,

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

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

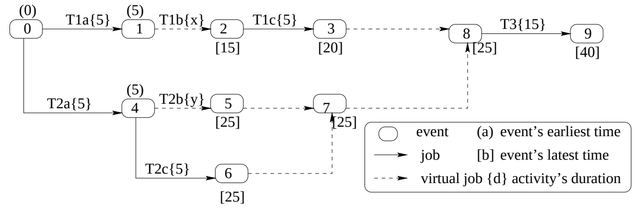

Figure 11: partial order schedule with two indirectly related issues

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.*

4.5 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. On high level it enables negotiation over intervals rather than fixed placement. 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 can only 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.

Figure 11 shows the partial order 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 to be finished between [0, 5]. Given each task has a duration of 5, it is impossible to find a feasible local schedule.

2. The two ranges [5, 15] for T1b and [10, 25] for T2b are consistent, which means that 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

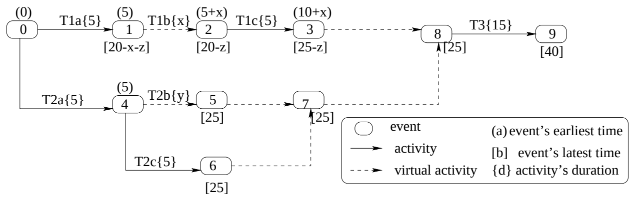

Figure 12: partial order schedule with two directly related issues

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 12 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 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 a 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 performed 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 continuous 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't 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 the 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 an 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.
3. Dynamic MQ. The agents negotiate about both 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 13 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.

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.

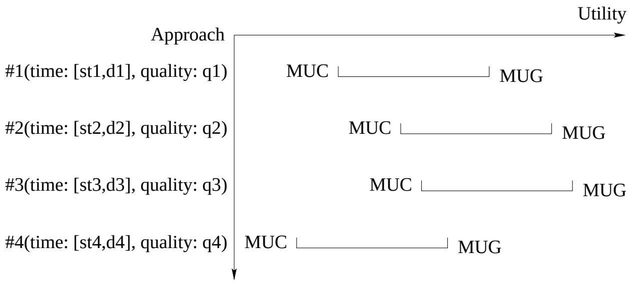

Figure 13: MQ level negotiation

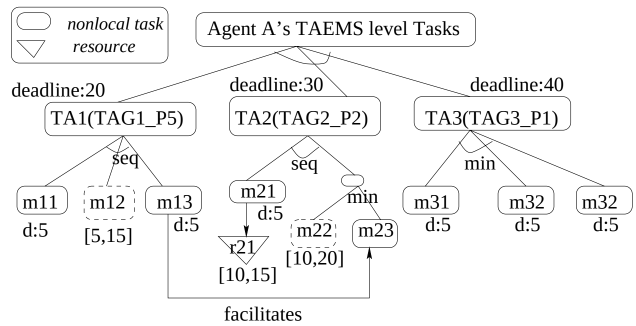

Figure 14: TÆMS level tasks

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 to contract out NL), the counter offer is acceptable; otherwise, it is rejected.

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, agent B can evaluate 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 the 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 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 if 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 predefined range. No matter what request agent A will make, or even if 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 is still 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.

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 14 shows agent A's current tasks and the required negotiation issues. Agent A currently has three tasks, task1, task2 and task3. All methods appearing 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.

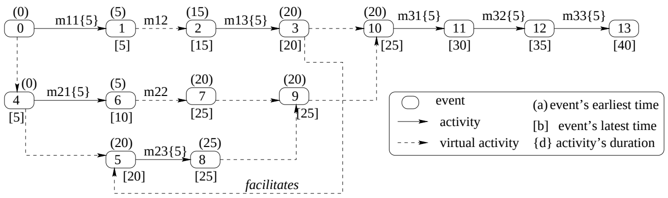

Figure 15: TÆMS level partial order schedule

Also there are two commitments built on MQ level for the non-local 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 belong 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 14 :

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 15).
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 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't 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 grant 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 a 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.

5 Conclusion

In this work we have defined a framework to study how the organizational relationships among agents (from cooperative to self-interested) affect the negotiation strategies and the influence on the performance of the multi-agent system. Also we show 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. A multi-leveled negotiation approach allows negotiation to be performed on different abstraction levels with different emphasis. Flexibility is introduced into negotiation and is been quantitatively reasoned about in the negotiation process. 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. Implementation process is in progress, we have built the basic agent architecture and the partial-order reasoning toolkit.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

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