A Multi-Leveled Negotiation Framework*

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

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

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

ABSTRACT

In this paper, we present a multi-leveled negotiation framework in which the negotiation process is performed at two 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 and resource-specific commitments among agents. The negotiation at this lower level involves the refinement of the rough commitments proposed at the upper level. The experimental work shows this two-leveled negotiation framework enables the agent to handle complicated negotiation issues and uncertainties in a more efficient way.

1. NEGOTIATION AT DIFFERENT LEVELS

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. This negotiation can require a complicated reasoning process when the agent has multiple tasks where the tasks may be achieved in different ways and include a sequence of activities, some of which may require external or internal resources. Additionally, uncertainty in task execution may further complicate the negotiation process as behavior deviates from the expected. The deviation can cause re-negotiation over commitments or the adjustment of local activities so as to still meet the commitments. This paper explores an alternative approach to negotiation in which the negotiation process is performed at different abstraction levels to reduce the complexity of the search. 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 and resource-specific commitments among agents. The negotiation at this lower level also involves the refinement of the rough commitments proposed at the upper level.

Let's look at an example to make these issues concrete. Agent α is Adam's personal assistant agent. Agent α is deigned to carry out multiple tasks corresponding to Adam's multiple goals in his life. Adam is a professor of Asian culture and language and he also has a family. He is asked by his department chair whether he can deliver a talk about his recent research results at the college. Also, he is planning to attend a conference in his research area. Meanwhile, his wife discusses with him the arrangement for their son's birthday party. Thus, there are three candidate tasks that appear in the agenda of agent α: prepare a talk for Adam's lecture, plan Adam's trip to a conference, and organize a birthday party for Adam's son. These tasks are associated with Adam's different roles, and contribute to different goals. The contributions of these tasks are not interchangeable. Each task has a deadline request, and also has multiple alternative ways to be performed. Figure 1 shows these three tasks. The higher level view describes the deadline for each task, the abstracted plans for each task, the duration of these plans and how they contribute to different goals. The lower level view describes the detailed plan for each task with the specification of the execution characteristics for each primitive tasks. Figure 2 presents the detailed plan for task *prepare talk*.

Agent α needs to make decisions about which tasks should be performed, and when and how to perform them. The possible issues that agent α can negotiate about include:

1. Negotiation with the secretary agent about when the talk should be delivered, which affects the deadline of the task *prepare talk*.
2. Negotiation with a translator agent about the task *translate material*, which includes when this task can be performed and how much it costs.
3. Negotiation with a travel agent about the task *book ticket*, which includes when this task can be performed and how

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

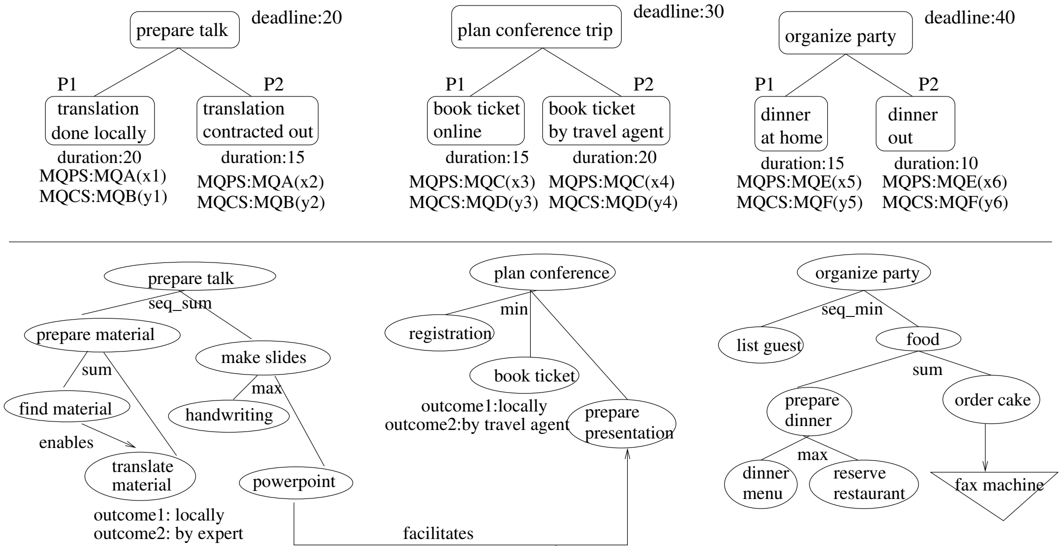

Figure 1: Agent A's three tasks

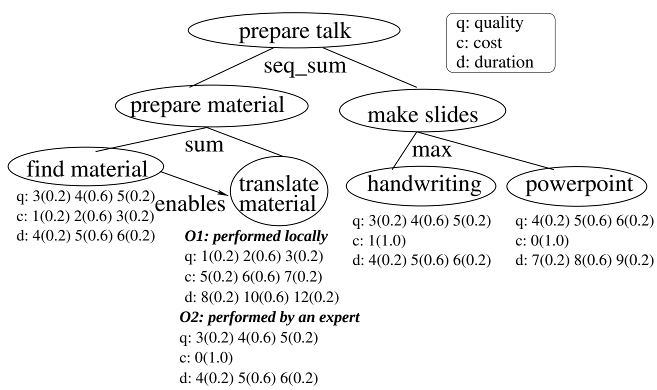

Figure 2: The detailed task structure of task "*prepare talk*"

much it costs.

4. Negotiation with agent β, the personal assistant agent of Adam's wife, about the task *organize party*, whether agent β can perform part of this task or the whole task.

It is reasonable to make these high-level decisions about whether to do these tasks locally or negotiate without a detailed model of the attributes associated with these tasks. All that needed is a rough view of the expected qualities of different ways to accomplish the task and their resource requirements. As in the early work in non-linear planning [2], it is important to leave flexibility in the higher level plan so that as more detailed constraints are introduced at lower level, there is room to accommodate them.

For example, agent α needs to perform task *prepare talk*, and there are two available plans for task *prepare talk* [1]:

1. P1: prepare the talk with the translation work done locally
2. P2: prepare the talk with the translation work contracted out to a translating agent

Each plan has different quality, duration and cost characteristics. The plan $P2$ requests contracting a subtask *translate material* to another agent. From the high level view, if agent α can find another agent to perform the subtask *translate material* before time 15 and with transferred utility less than 5, the plan $P2$ is the best choice. The availability of this commitment affects agent α's local plan. If such a commitment is not available, agent α needs to choose the other plan $P1$ for task *prepare talk*. The other plan $P1$ takes a longer time for execution and hence makes it impossible for another task *plan conference trip*, which agent α has planned to accomplish, to meet its deadline. Agent α has to change its current local plan. By comparing its original local plan with the commitment on *translate material* and the other plan without the commitment on *translate material*, agent α can find out how important it is to obtain a commitment on *translate material*.

However, not all issues can 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 about 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, 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 obtained from another agent), should be negotiated first at the upper level and rough commitments should be constructed for them. However, we argue that those issues which have less influence on local plan selection and involve reasoning about the detailed structure of the low level activities, do not have to be directly reasoned 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 *powerpoint* (make slides using powerpoint) that belongs to *prepare talk* facilitates the subtask *prepare presentation* that belongs to *plan conference trip* because part of the slides for the lecture can be reused in the conference presentation if the slides are done in powerpoint format. This relationship is not visible from the high level tasks. Besides, whether the subtask *powerpoint* is included in the plan for task *prepare talk* depends on which plan is selected for this task at the higher level reasoning process. However, the agent can exploit it to optimize its local plan after the high level plan is decided.

[1] To make the example simple, only two plans for each task are shown in Figure 1, although there are other plans for this task besides these two plans.

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. For example, the higher level plan $P2$ for task *prepare talk* has an estimated duration of 15, which is based on the expected value of the primitive tasks' durations. Figure 2 shows the uncertainty information for each primitive tasks.
3. *Internal resource requirement associated with low level tasks.* For example, agent α needs to use the fax machine for task *order cake* (Figure 1), but it shares the fax machine with several other agents. Given the knowledge of the general usage of fax machine, the agent knows it is unnecessary to reserve the fax machine when it builds its high level plan. But when the agent comes to arrange its local activities, it should consider this resource constraint.

Considering the above issues, the agent may need to revise its higher level commitments through the lower level negotiation and additionally to reorder its lower level activities, so as to optimize it local plan and commitments, reduce failure possibilities, avoid conflicts and achieve higher utilities.

A multi-leveled negotiation framework is introduced in this paper. First we will present the supportive frameworks in Section 2. Then we will describe the overview ideas of the multi-leveled negotiation in Section 3. Examples are used to explain how this framework works in Section 4. Different reward models are discussed in Section *4.3.1*. Section 5 shows how these different reward models affect the agent's performance. Section 6 summarizes this paper.

2. SUPPORTIVE FRAMEWORKS

The multi-leveled negotiation is performed at different abstraction levels. In this work, the MQ framework [4] is used for the higher level representation, while the TÆMS framework [1] is used to support the lower level reasoning process. However, the basic approach is not restricted to these two frameworks, and we feel they can also be applied to other suitable task representation frameworks.

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 motivated by self-interested reasons to cooperative reasons. Each agent has a set of MQs or motivational quantities that it tracks and accumulates. *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 the MQ *production set* ($MQPS$), and consumes some quantity of MQs, called the MQ *consumption set* ($MQCS$).

The TÆMS language [1] (See Figure 5) is a domain-indep-endent task modeling language, which allows us to model agent's detailed activities. The agent's candidate tasks are described in hierarchical structures with alternative ways of accomplishing tasks. The primitive tasks (methods) are characterized by three features: quality (q), duration (d) and cost (c) via discrete probability distributions. 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. Resource requirements of methods are also explicitly modeled in TÆMS framework.

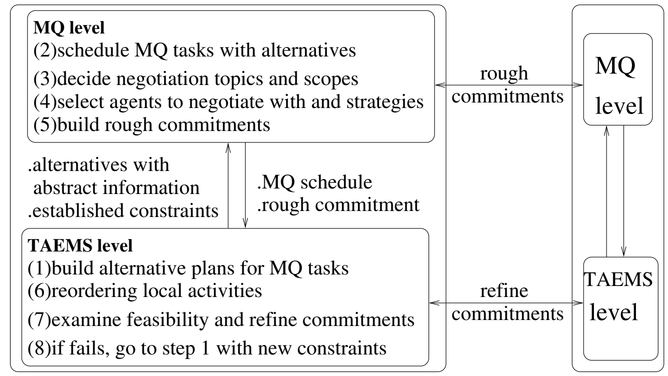

Figure 3: two-level negotiation framework

3. OVERVIEW OF BASIC IDEAS

The MQ model [4] describes the agent's organization knowledge about task utility but it does not support detailed model of tasks and their interactions, and lacks of representation of the uncertainty characteristics and resource requirements of tasks, which belong to the TÆMS [1] model. The proper integration of the reasoning processes that operate on these different task models enables agents not only to reason about the organizational concerns but also to handle detailed feasibility analysis and implementation of tasks.

An agent has its MQ level view of its local activities, which is a set of potential MQ tasks, each associated with certain $MQPS$ and $MQCS$, which can be mapped into the agent's utility given the agent's current MQ state. For example, Figure 4 shows that agent A has three MQ tasks, $T1$, $T2$ and $T3$. $T1$ 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 $T1$. For each MQ task T, there is a TÆMS task structure that describes the detailed activities for this task, i.e. the task structure $TG1$ in Figure 5 describes the detailed activities in task $T1$. Different plans to accomplish the MQ task T can be generated from the TÆMS task group TG by the DTC scheduler, and each plan has different quality, duration and cost characteristics that affect the $MQPS$ and $MQCS$ of the task T (See Figure 6). This is the first step [step 1] shown in Figure 3, 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 partial order schedule allows future refinement at the lower level where more detailed information is available. 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

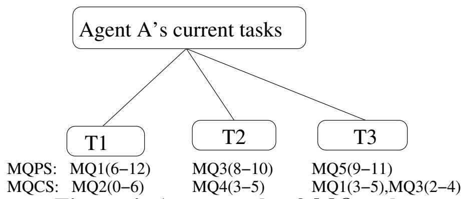

Figure 4: An example of MQ tasks

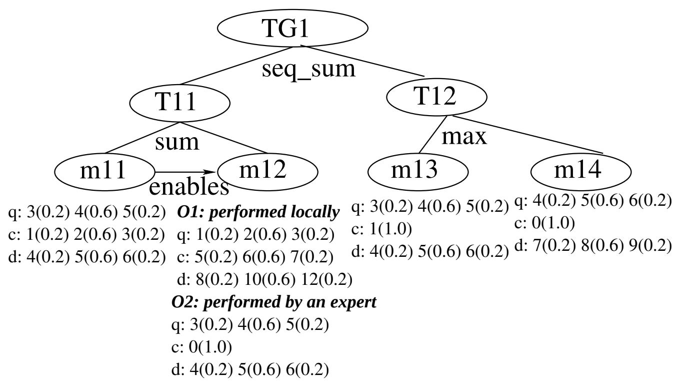

Figure 5: Task structure for T1

another agent, or receiving 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 can select which agents to negotiate with and the appropriate negotiation strategy according to organizational relationships and the negotiation issues [step 4]. The MQ level negotiation builds rough (partial-specified) commitments for those issues that should or could be reasoned about 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 optimizes its local schedule by taking advantage of the interrelationships among low-level tasks/methods. Also the agent verifies 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 used to manage and reason about these relationships and constraints on the TÆMS level. Negotiation on the TÆMS level involves refining those rough commitments as needed when:

1. There are conflicts or potential conflicts among commitments and local activities caused by additional constraints (such as a local resource constraint) or uncertainties in real-time execution.
2. It is possible to reduce local cost or increase local utility by refining a commitment.

If the agent can find a feasible local schedule by reordering and renegotiation on the TÆMS level, it can execute its local schedule and perform all of its commitments. If unexpected events cause conflict in the execution process, the agent needs to check if the conflict can be solved by refining any commitments. Otherwise, if the conflict can't be resolved given all current constraints, the agent needs to discard some commitments (decommits), establish other

name	plan	q	c	d
TG1_P1	(m11, m12, m13)	10	9	20
TG1_P2	(m11, m12, m14)	11	8	23
TG1_P3	(m11, m13)	8	3	10
TG1_P4	(m11, m14)	8	2	13
TG1_P5	(m11, $[m12]^2$, m13)	12	3	15
TG1_P6	(m11, [m12], m14)	13	2	18

commitments on already scheduled local activities and go back to the MQ level to reschedule, and possibly result in constructing new commitments [step 8].

4. THROUGH THE PROCESS

In this section, we will discuss this two-level negotiation in more detail using examples.

4.1 DTC Scheduler Builds Alternatives

The Design-To-Criteria (DTC) scheduler [3] is a domain-independent scheduler that aims to find a feasible schedule that matches the agent's particular criteria request. In this research, it is 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 $T1$, $T2$ and $T3$, which are mapped into the task groups $TG1$, $TG2$ and $TG3$ in the TÆMS model. There is a subtask $m12$ of $TG1$ (See Figure 5) that potentially can be contracted to another agent who is an expert on task $m12$.

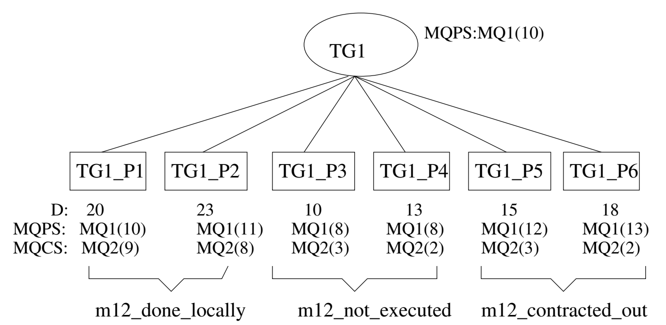

Figure 6: Task $T1$'s alternatives

The DTC scheduler works on $TG1$ according to the following different assumptions:

1. $m12$ is executed locally;
2. $m12$ is not executed;
3. $m12$ is contracted to another agent.

These assumptions can be combined with different quality, cost, and duration scheduling criteria to generate the following set of alternative plans:

Each plan has different performance characteristics, corresponding to an MQ level alternative with different duration, $MQPS$, and $MQCS$, as shown in Figure 6. In this example, the following functions describe how the quality and cost characteristics of a plan P_n are mapped into the $MQPS$ and $MQCS$, for task $T1$:

$$MQPS : MQ1(P_n) = quality(P_n)$$
$$MQCS : MQ2(P_n) = cost(P_n)$$

This is a simple example of the mapping function. However, the mapping function could be more complex using more features such as: the likelihood of meeting the deadline, the maximum derived quality rather than the expected, etc. The structure of the function also depends on the problem-solving context.

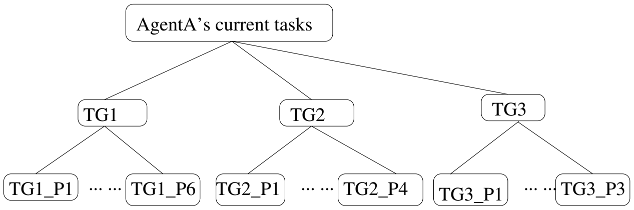

Figure 7: MQ level tasks

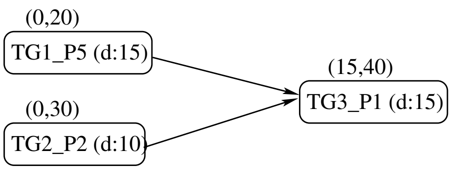

Figure 8: MQ level partial order schedule

For those plans that need to contract $m12$ to another agent, such as $TG1\_P5$ and $TG1\_P6$, the MQCS does not include the cost for contracting the task $m12$, because the cost is unknown at this time. Similarly, different plans are generated for task $T2$ and $T3$. This abstraction process can be done off-line, and these alternative plans can be stored in the agent's database.

4.2 MQ Level Scheduling

The MQ level scheduler does scheduling for these alternatives of $T1$, $T2$ and $T3$ to find the best schedule MQ_S1 that provides the agent the most utility increase from its current state (Figure 7). If the plan $TG1\_P5$ or $TG1\_P6$ ($m12$ is contracted out) appears in the scheduler MQ_S1, agent A needs to consider contracting $m12$ to another agent; otherwise, agent A may choose to execute $m12$ locally or not to perform $m12$ as the schedule MQ_S1 recommends. Suppose the best schedule MQ_S1 includes the $TG1\_P5$ plan:

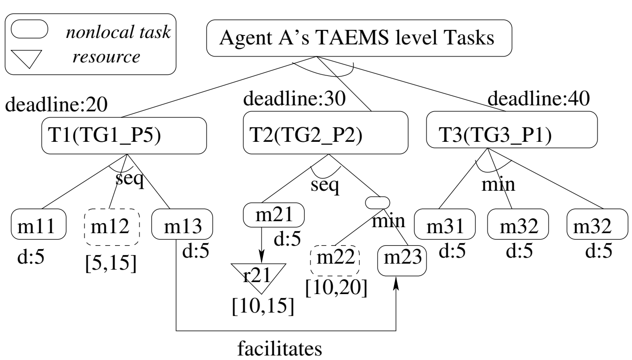

Figure 9: TÆMS level tasks

$TG1\_P5$[duration:[3] 15 earliest start time:0 deadline:20]
$TG2\_P2$[duration:10 earliest start time:0 deadline:30]
$TG3\_P1$[duration:15 earliest start time:10 deadline:40]

This is a partial order schedule (See Figure 8). $TG1\_P5$ and $TG2\_P2$ need to be finished before $TG3\_P1$ starts. The reason is that $TG3\_P1$ consumes the MQs produced by $TG1\_P5$ and $TG2\_P2$.

Agent A compares the utility of the best schedule including the contracting plan of $m12$ (MQ_S1) with the utility of the best schedule without the contracting plan of $m12$ (MQ_S2):
$TG1\_P4$[duration:13 earliest start time:0 deadline:20]
$TG2\_P2$[duration:10 earliest start time:0 deadline:30]
$TG3\_P1$[duration:15 earliest start time:10 deadline:40]

The difference is the utility gained by contracting $m12$ to another agent.

Marginal_Utility_Gain($m12$) = Utility(MQ_S1) - Utility(MQ_S2)

Marginal utility gain specifies the local utility increment by contracting this task to another agent. On the other hand, *marginal utility cost* specifies the local utility decrement for the contractor agent by performing this task without considering the potential benefits the contractor agent can get from the transferred MQ with the task. These two measures are used by the agents to guide the negotiation on the transferrred MQ[7].

The basic constraint of the quality request and the temporal constraint of $m12$ is established based on the TÆMS level schedule ($TG1\_P5$) and the MQ schedule (MQ_S1). Suppose in the $TG1\_P5$ schedule, the quality request of $m12$ is 10, and the abstraction of the schedule $TG1\_P5$ is (5, $m12$, 5); it means there are some activities of duration 5 that need to be done before $m12$ and some activities of duration 5 that need to be done after $m12$. Combined with the temporal constraint of $TG1\_P5$ in the schedule MQ_S1 [0, 20], the temporal scope of $m12$ is [5, 15].

Agent A posts this task allocation proposal as:
$m12, quality - request : 10, time - scope : [5, 15]$

4.3 MQ Level Negotiation

The negotiation on the MQ level includes the following concerns:

1. For each issue in negotiation, there are multiple features could be negotiated about, such as the transferred MQ, the different approaches of the task and the reward model. The negotiation is multi-dimensional.
2. For each negotiation session, there are different negotiation protocols available, such as single step negotiation or multi-step negotiation. The agent needs to find the appropriate negotiation protocol.
3. Although there is only one non-local tasks in negotiation in this example, it is often the case that there are multiple issues in negotiation and the negotiation on one issue affects the negotiations on other issues. The agent needs to decide the ordering of these negotiations and how it should negotiate on each issues.
4. Give the other agents in negotiation may have different organizational relationships with this agent, the agent needs to choose appropriate negotiation attitudes toward other agents. This problem can be addressed by introducing the *relational MQ* which represents the relationship between agents.

The above problems have been studied as multi-dimensional negotiation, multi-step negotiation, multi-linked negotiation and integrative negotiation. The details are presented in [8, 6, 7], Those

[3] The duration of a plan describes how long it takes to execute this plan. It includes both the local process time and the non-local process time. For plan $TG1\_P5$, there are 5 time units for $m12$ which is non-local process time because $m12$ is executed by another agent. It is possible for agent A to arrange other local tasks in this period of time. However, the MQ scheduler does not reason at this level of detail. The partial order scheduler which works at the TÆMS level, checks all local activities and makes sure the local process time is not wasted.

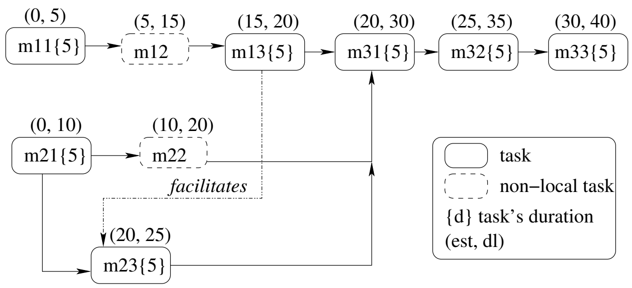

Figure 10: TÆMS level partial order schedule

approaches all fits into this multi-leveled negotiation framework. In this paper, we only describe how the agent selects an appropriate reward model that takes into account the possible further refinement of the rough commitment. Agents build rough commitments as a result of the MQ level negotiation. Future refinement as result of the lower level (TÆMS) negotiation is possible given the range specified by the rough commitment. The refinement will affect the flexibility of the commitment and hence affect the value/cost of the commitment. Thus agents need to negotiate over the reward model which specifies how the refinement is related to the value of the transferred MQ. Since the reward model is related to the negotiation on both levels, we will discuss this in detail.

4.3.1 Reward Models

Agents build rough commitments as a result of MQ level negotiation. They are rough commitments since the specifications can be ranges rather than points; these ranges allow further refinement. For example, a rough commitment c could specify the temporal constraint for the contracted task NL to be started and completed somewhere between $[t1, t2]$. If $F(c) > 0, t2 > t1 + d$ ($F(c)$ denotes the flexibility of c; d denotes the estimated 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 flexibility* model. The contractee agent E pays $v1$ of MQ_i for the contractor agent R to perform task NL during any time period (not shorter than d) within $[t1, t2]$ as agent E requests. This agreement provides agent E with the freedom to further refine this commitment, and agent R agrees to accommodate any request from agent E within the predefined range. No matter what request agent E will make, or even if agent E does not make any further requests, agent R will receive $v1$ of MQ_i as decided in the rough commitment.
2. *Dynamic flexibility* model. The contractee agent E pays $v2$ of MQ_i for the contractor agent R to perform task NL within the range of $[t1, t2]$. If agent E requests a restriction on this range to $[t1 + x, t2 - y], (t2 - y - t1 - x \geq d)$ and if agent R could accept this request, agent E will pay $((x + y) * \beta + 1) * v2$ of MQ_i to agent B. Agent B would decide to accept this additional refinement request or not, according to its current problem-solving context. If agent R does not accept this request, it is still obliged to perform NL during $[t1, t2]$ and in turn is guaranteed to get $v2$ of MQ_i 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.

4.3.2 Uncertainties

The uncertainty discussed here refers 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:

$V : \{v_1(p_1); v_2(p_2); ...; v_n(p_n)\}$

meaning variable V has chance of p_i of being 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 :

$$MU(V) = -\sum_i p_i * log(p_i) * \frac{|v_i - E(V)|}{E(V)}$$

Probability of Above Expectation of variable V :

$$PAE(V) = \sum_{i|v_i > E(V)} p_i$$

Measure of Above Uncertainty of variable V :

$$MAU(V) = \sum_{i|v_i > E(V)} p_i * (v_i - E(V))$$

For example:

$X = \{3(0.3); 5(0.4); 7(0.3)\}$;
$E(Y) = 5$; $MU(X) = 0.125$; $PAE(X) = 0.3$; $MAU(X) = 0.6$;
$Y = \{4(0.1); 5(0.8); 6(0.1)\}$;
$E(Y) = 5$; $MU(Y) = 0.04$; $PAE(Y) = 0.1$; $MAU(Y) = 0.1$;

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

4.3.3 Reasoning about Uncertainty

The general approach to accommodate uncertainty in this negotiation framework is described as follows. In the low level reasoning process, uncertainties are represented as statistical distributions[4]. Uncertainty information is abstracted as the expected value, the marginal value, the *probability of above expectation* and the *measure of above uncertainty* as described above. This abstracted information is used in the upper level reasoning process. The upper level process does not deal with the detailed distribution information. Given the marginal value and the probability of the above expectation, the agent chooses the appropriate reward model. If the *probability of the above expectation* is large (bigger than a pre-set limit) or the *measure of the above uncertainty* is large, the agent chooses the *pre_paid flexibility* model. Otherwise it chooses the *dynamic flexibility* model. The marginal value is attached to the commitment to describe that a specified item in this commitment may need to be changed by the extent of the marginal value. If the contractee agent promises to accommodate this change when requested by the contractor agent (*pre_paid flexibility* model), it can charge a higher price for this commitment but it also needs to reserve enough room in its local plan for the future change. Otherwise, the contractee agent can choose the *dynamic flexibility* model. In this way it does not promise to accommodate the future change. When the contractor agent requests a change, it checks its local plan to see if this change can be guaranteed. If so, an extra cost is added when the change really happens.

[4]The distribution explosion problem is handled using the clustering method and in the clustering process, the marginal values are preserved[5].

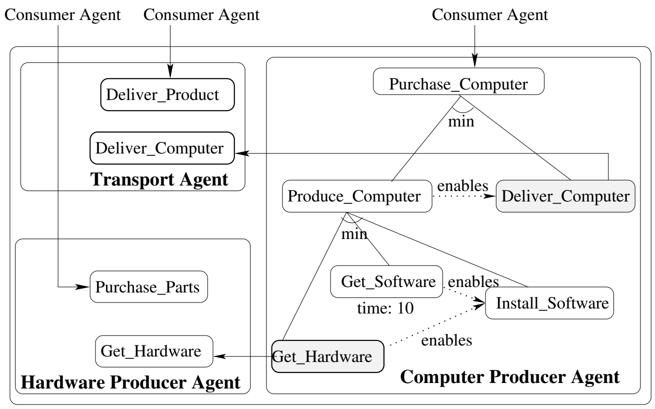

Figure 11: Supply Chain Example

4.4 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 9 shows agent A's current tasks and the required negotiation issues. Agent A currently has three tasks, $T1$, $T2$ and $T3$. All methods appearing in this figure are those constructing the plan $TG1\_P5$, $TG2\_P2$ and $TG3\_P1$. $T1$ has a deadline of 20; $T2$ has a deadline of 30, and $T3$ has a deadline 40. $T1$ and $T2$ need to be finished before $T3$ starts. These constraints come from the MQ level scheduling. Also there are two commitments built at the 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 other high-level tasks: they are not visible to the MQ level scheduler so they are not reflected in the MQ level schedule. Two examples are shown in Figure 9:

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 10).
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 interrelationships 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 agents A and C is the *pre-paid flexibility* model, then agent C would accept this request. Otherwise, if the commitment is associated with the *dynamic flexibility* model, agent C needs to reason about its local partial order schedule to determine if it can grant this request. If it can, agent C will get extra MQ from agent A as they have agreed on at the MQ level negotiation. If this refinement negotiation is successful, agent A can generate a new 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]

Besides the additional constraints caused by resource requirements and the relationships among those subtasks that belong to different high-level tasks, the other reason for TÆMS level negotiation is the uncertainty of task execution. More details about the uncertainty will be discussed in Section 4.3.2.

4.5 MQ Level Rescheduling

If the refinement negotiation fails and agent A can not find a feasible local schedule given all local constraints, agent A has the following choices:

1. Select a similar plan with a different schedule and try to solve the conflict.
2. Discard some impossible tasks/commitments.
3. Reschedule at the MQ level, given the current commitments, tasks, and newly arrived tasks.

The first two choices cause the agent to generate a schedule which is different from the original one that was optimal given the knowledge at the time of scheduling; hence the agent's utility achievement won't be as good as it expects. However, the choice of rescheduling on the MQ level may involve much higher cost compared to the first two choices, although it promises to provide an optimal solution given all current knowledge. So the agent needs to compare the loss of utility as a result of following a sub-optimal solution to the cost of rescheduling.

5. EXPERIMENTAL WORK

The experimental work in this section studies how the two-leveled negotiation mechanism affects the agent's performance compared to an one-leveled negotiation. We further study how the upper level negotiation (the choice of reward model) affects the lower level negotiation and hence affects the agent's performance.

The experiments use the same supply-chain example described in Figure 11. *Consumer Agent* generates orders such as *Purchase_Computer*, *Purchase_Parts* and *Deliver_Product* for different agents including *Computer Producer Agent*, *Hardware Producer Agent* and *Transport Agent*. In order to accomplish task *Produce_Computer*, *Computer Producer Agent* needs to generate an external request for hardware (*Get_Hardware*), and also needs to deliver the computer (*Deliver_Computer*) through a transport agent. The agents negotiate about these orders. New tasks were randomly generated. Uncertainties are introduced by the execution component which generates the execution time for a task according to its statistical distribution. This scenario represents a class of problems with real-time uncertainties on tasks' execution times, where some of the commitments may be changed to avoid the missing of deadlines. If a task takes longer than the expected time, it may cause other tasks to miss their deadlines. The lower level negotiation occurs when this delay can be avoided by refining some rough commitments of non-local tasks. The other reasons for lower-level negotiation, such as additional constraints caused by resource requirements or reordering of the lower level tasks, didn't occur in this experimental setup; however, the two-leveled negotiation mechanism is capable of supporting re-negotiation caused by all types of reasons.

Four different policies are tested:

1. Fixed policy: The commitment built on the upper level (MQ level) is fixed; there is no lower level re-negotiation to refine the commitment from the upper level.

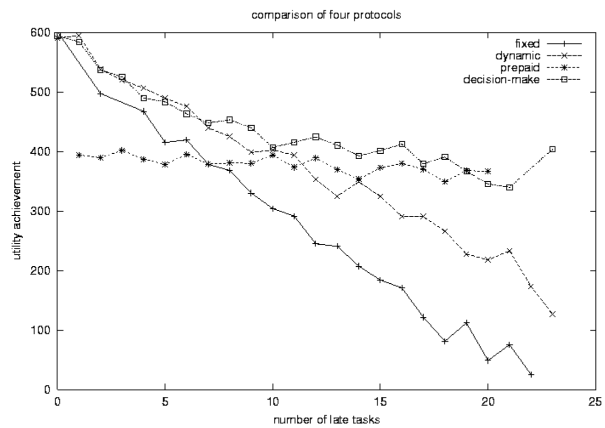

Figure 12: ***Computer_Producer_Agent*'s performance using different policies when uncertainty changes**

2. Dynamic flexibility policy: The agent always chooses the *dynamic flexibility* reward model in the upper level negotiation.
3. Pre-paid flexibility policy: The agent always chooses the *pre-paid flexibility* reward model in the upper level negotiation.
4. Decision-making flexibility policy: In the upper level negotiation, the agent chooses either the *dynamic flexibility* reward model or the *pre-paid flexibility* reward model according to the abstracted uncertainty information, as described in Section *4.3.3*.

The entire experiment contains 225 group experiments. Each group experiment has the system running for 1000 time clicks for four times and each time the agents use one of the four different strategies. We focus on the performance of *Computer_Producer_Agent* because it is the only agent who needs to sub-contract its subtask to other agents.

Figure 12 shows that when uncertainty increases (the number of late tasks increases), the agent's performance decreases significantly without the lower level negotiation (using the fixed policy). The reason is that the agent can not get the expected reward without finishing the task on time; additionally it has to pay the decommitment penalty. The lower level negotiation helps the agent to adjust its previous commitment with the other agent, so as to accommodate the uncertainties and avoid missing tasks' deadlines. As the uncertainty increases, the performance of the dynamic flexibility policy decreases, because the dynamic flexibility policy can not guarantee the success of the lower level negotiation. The other agent may accept the adjust request or not depending on its current problem solving context. With the pre-paid flexibility policy, the agent's performance is almost stable regardless of the change of the uncertainty. The agent always pre-pays for the flexibility to adjust the rough commitment whether it needs it or not. When the uncertainty is lower (the number of late tasks is less than 9), the agent actually wastes some of its potential gain by paying for flexibility it does not need. The decision-making flexibility policy brings the agent the nearly-best performance in all situations with different uncertainties, because the agent can reason about when it may need flexibility and can pre-pay for it, or when it may not need extra flexibility and can save money on the contract.

6. SUMMARY

One major difference between this work and other work on negotiation is that it is not viewed as a stand-alone process. Rather, it is viewed as one part of the agent's activity, which is tightly interleaved with the planning, scheduling and execution of the agent's activities, including other negotiations. This recognition has led us to a multi-level negotiation framework that allows us to handle the complexity inherent in this view. In this framework, an agent reasons about and negotiates over more important issues at the upper level (MQ level), and then refines the rough commitments at the lower level in order to optimize its local plan and accommodate additional constraints and uncertainties. Examples are used to explain how a number of different technologies, such as MQ, TÆMS and DTC can be incorporated to support sophisticated negotiation. The multi-linked [6] and integrative negotiation[8] technologies that are described in other papers can also fit into this framework. Additionally, agents can choose an appropriate reward model in the higher-level negotiation according to the uncertainty measure; hence, the agent can pay for its local flexibility to accommodate the future uncertainty. The two-leveled negotiation framework enables the agent to reason about complicated negotiation issues and uncertainties in a more modular and computationally efficient manner. It also allows the agent to reason about the organizational concerns, implementation of objectives, and negotiation and re-negotiation decisions in an integrated way. This architecture opens up a wide variety of future work directions. We are especially excited about the potential of its use for studying agent behavior in a complex organizational context.

7. REFERENCES

[1] K. S. Decker and V. R. Lesser. Quantitative modeling of complex environments. *International Journal of Intelligent Systems in Accounting, Finance, and Management*, 2(4):215–234, Dec. 1993.

[2] E. Sacerdoti. The nonlinear nature of plans. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*.

[3] T. Wagner, A. Garvey, and V. Lesser. Criteria-Directed Heuristic Task Scheduling. *International Journal of Approximate Reasoning, Special Issue on Scheduling*, 19(1-2):91–118, 1998.

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

[5] X. Zhang. *Sophisticated Negotiation In Multi-Agent Systems*. PhD thesis, University of Massachusetts Amherst, 2002.

[6] X. Zhang and V. Lesser. Multi-linked negotiation in multi-agent system. In *Proceedings of the First International Joint Conference on Autonomous Agents And MultiAgent Systems (AAMAS 2002)*.

[7] X. Zhang, V. Lesser, and R. Podorozhny. New results on cooperative, multistep negotiation over a multi-dimensional utility function. In *AAAI Fall 2001 Symposium on "Negotiation Methods for Autonomous Agents,"*.

[8] X. Zhang, V. Lesser, and T. Wagner. How cooperative should i be? In *Proceedings of the First International Joint Conference on Autonomous Agents And MultiAgent Systems (AAMAS 2002)*.