A Layered Approach to Complex Negotiations*

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

May 17, 2004

Abstract

In this paper, we present a layered agent framework in which the negotiation process is performed at different levels of abstractions. In a real time, multiple-tasking, resource-bounded environment, negotiation is not an isolated process but one that interleaved with agent's many other activities, such as scheduling, execution, and other negotiations. To make the complexities of negotiation more tractable, the negotiation process is performed at two 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, in what order the multiple related negotiations should be performed, and what negotiation attitude should be used for each negotiation. 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; negotiation at this level involves refinement of the rough commitments proposed at the upper level. The experimental work shows this two-level negotiation framework enables the agent to handle complicated negotiation issues and uncertainties in a more efficient way.

Keyword: layered negotiation, autonomous agents, multi-agent systems

1 Introduction

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. However, for complex agent-based applications operating in dynamic, open environments, the agents may have multiple and complicated tasks; each task may be achieved in different ways and include a sequence of activities, some of which may require external or internal resources. The agent needs to choose which tasks to perform, when to perform them and how to perform them. The successful execution of a task may involve negotiation with other agents about sub-contracts or resource requirements. Meanwhile, since the agent works in a complex organizational context, it needs to work with other agents from a variety of different organizational positions. Hence the negotiation attitude should conform to the organizational relationship. 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. Given all the above considerations, it is difficult to construct an integrated framework in which all these issues are addressed concurrently and done so in an efficient way. One major difference between this work and other work in negotiation is that in this work negotiation is not viewed as a stand-alone process. Rather, it is viewed as one of an agent's many interleaved activities – including scheduling, execution, and other negotiations. This view plus the complexities of negotiation mentioned earlier has led us to construct a two-level framework that makes the complexity inherent in this view more tractable. In this two-level negotiation framework, the negotiation process is performed at different abstraction levels to reduce the complexity of the search. An agent thus reasons about and negotiates over more important issues at the upper level, and then refines the rough commitments at the lower level in order to optimize its local plan and accommodate

*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.

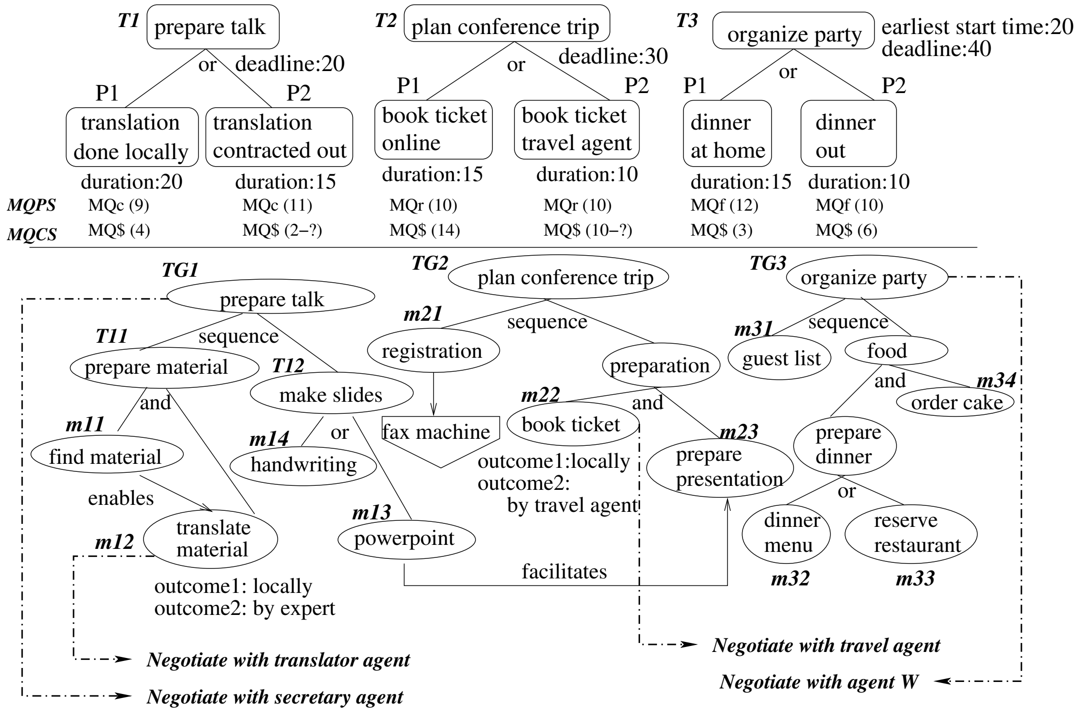

Figure 1: Agent A's three tasks

additional constraints and uncertainties. The focus of this work is on the decision-making process of negotiation, rather than the negotiation protocol or the language.

Let's look at an example to make these issues concrete. Agent A is Adam's personal assistant agent. Agent A is designed 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. His department chair asks him whether he can deliver a college talk about his recent research activities, which requires some foreign material being translated. This task contributes to the goal of serving the college, represented by generating certain amount of MQ_c (See Section 2 for explanation of MQ). At the same time, he is planning to attend a research conference. This task contributes to the goal of academic research, represented by generating certain amount of MQ_r. Meanwhile, his wife discusses with him the arrangement for their son's birthday party. This task contributes to the goal of serving the family, represented by generating certain amount of MQ_f. Thus, there are three candidate tasks that appear in the agenda of agent A: prepare a talk for Adam's college 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 has multiple alternative ways for it to be performed. Figure 1 shows these three tasks. The upper-level view describes the temporal constraints (including the the earliest start time and the deadline) for each task, the abstracted plans for each task, the duration of these plans and how they contribute to different goals (in terms of generating different type and amount of motivational quantities (MQ)). For example, there are two different plans to accomplish task *prepare talk*: plan P1 is to prepare the talk with doing the translation work locally, plan P2 is to prepare the talk with contracting the translation work to other agent. The lower-level view describes the detailed task structure for each task, specifies the execution characteristics for primitive tasks and the relationships among these primitive tasks. For example, TG1 describes the detailed task structures for task T1 *prepare talk*: to prepare talk, two subtasks, *prepare material* and *make slides*, need to be performed in sequence. To *prepare material* , the agent needs to first *find material* and then *translate material*. There are two different approaches to translate material, either to perform the translation work locally or to contract the translation work out. To *make slides*, the agent can either use *handwriting* or use *powerpoint*. There are more detailed descriptions of these representations on different abstract levels in Section 2.

In our example, agent A needs to make decisions about which tasks should be performed, when, and how to perform them (which alternative to choose). The possible negotiations (shown in Figure 1) that agent A may be involved include:

1. Negotiation with the secretary agent about when the college talk should be delivered. This 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 much it

costs.

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

These negotiation issues are inter-related, so called "multi-linked" negotiation. The result of one negotiation affects the other negotiations. Also the negotiations with different agents involves different organizational relationships, hence requires appropriate negotiation attitudes. The previously presented mechanism for multi-linked negotiation [14] and integrative negotiation [17] can be applied to this problem given the following architecture and process, we will discuss these issues in detail in Section 4.3.

In this paper, we take the position that it is reasonable to make high-level decisions about whether to (attempt to) perform tasks locally, or to negotiate over the tasks, without a detailed model of task attributes. All that needed is a rough view of the expected qualities of the different tasks, the expected qualities of alternative ways to perform the tasks, and any associated resource requirements. As in the early work in nonlinear planning [8], 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, as shown in Figure 1 agent A needs to perform task *prepare talk*, and there are two available high-level plans for the *prepare talk* task:

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[1] 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 A can find another agent to perform the subtask *translate material* before time 15 and with transferred utility less than 5, then plan $P2$ is the best choice. The availability of this commitment affects agent A's local plan. If such a commitment is not available, agent A needs to choose the other plan, $P1$, for task *prepare talk*. $P1$ takes longer to perform and hence makes it impossible for A to perform *organize party* using plan P1 (dinner at home) by its deadline. By comparing these two schedules – the one with the commitment on *translate material* to the other local schedule without the commitment on *translate material*, agent A can determine how important it is to obtain a commitment on *translate material* and perform $P2$ instead of $P1$.

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 decision-making in negotiation: 1) 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. 2) However, we argue that those issues which have less influence on local plan selection and involve reasoning about the detailed structure of the lower-level activities, do not have to be directly reasoned about 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 higher-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 schedule after the higher-level schedule 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 (or other abstraction model, see Section 4.3.2) are used in the higher-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 in making a commitment. The lower level has detailed information about the uncertainty of task execution, and since more context knowledge is available along with the process, the higher-level commitment can be adjusted to accommodate for uncertainty. 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 4 shows the uncertainty information for each primitive task.

3. *Internal resource requirements associated with lower-level tasks.* For example, agent A needs to use the fax machine for task *registration* (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 that it is unnecessary to reserve the fax machine when it builds its higher-level schedule. But when the agent comes to arrange its local activities, it should consider this resource constraint.

---

[1] Planning from first principles is not addressed in this paper. The term "plan" here indicates a set of selected and ordered activities generated by the scheduler from a set of candidate task structures – structures which identify the alternative ways that a task might be performed and their respective performance characteristics. The scheduler handles the choice and the sequencing of tasks.

Considering the above issues, the agent may need to revise its higher-level commitments through the lower-level negotiation. The agent may also have to reorder its lower-level activities, so as to optimize its local schedule and commitments, reduce failure possibilities, avoid conflicts and achieve higher utilities. A two-level negotiation framework is introduced in this paper. First we will present the supportive frameworks in Section 2, then we describe the basic underlying analytical ideas of the two-level negotiation framework in Section 3. Examples are used to explain how the framework operates in Section 4. Different reward models are discussed in Section 4.3. Section 5 shows how these different reward models affect the agent's performance. Section 6 presents more issues in MQ level negotiation, and Section 7 summarizes this paper and discusses related work.

2 Supportive Frameworks

The multi-leveled negotiation is performed at different abstraction levels. In this work, the MQ framework [13] is used for the higher-level representation, while the TÆMS framework [2] 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, as long as the upper-level framework provides goal-related representation and quantitively reasoning of utility and temporal constraints, the lower level framework models detailed task structures and associated uncertainty and resource information.

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. MQs represent progress toward organizational goals and in certain cases may be used as a medium of exchange. For example, in Figure 1, MQ_c, MQ_r, and MQ_f represent the progress towards the college service, the academic research, and the family service respectively. $MQ_\$$ represents the monetary accumulation, which can 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. In the example shown in Figure 1, task T2 *plan conference trip* produces some MQ_r, and also consumes some $MQ_\$$. 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.

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 (different plans), 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 task modeling language [2] (See Figure 4) is a domain-independent task modeling language. 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, duration and cost via discrete probability distributions. Quality describes the contribution of a particular method to overall problem solving. It is a domain-independent concept. Different applications have different notions of what the concept of quality models. Duration describes the amount of time that the method will take to execute, and cost describes the financial or opportunity cost associated with the performing of this action. The qaf (quality accumulation function) associated with each task describes how the qualities of its sub-tasks contribute to the quality of this task.

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.

The MQ model [13] describes the agent's organizational knowledge about task utility but it does not support detailed modeling of tasks and their interactions, and lacks of representation of the uncertainty characteristics and resource requirements of tasks. These details are represented using the TÆMS [2] task modeling language. The proper integration of the MQ and TÆMS models and reasoning processes enables agents to reason about both organizational level task value and to handle detailed feasibility, analysis, and implementation of tasks.

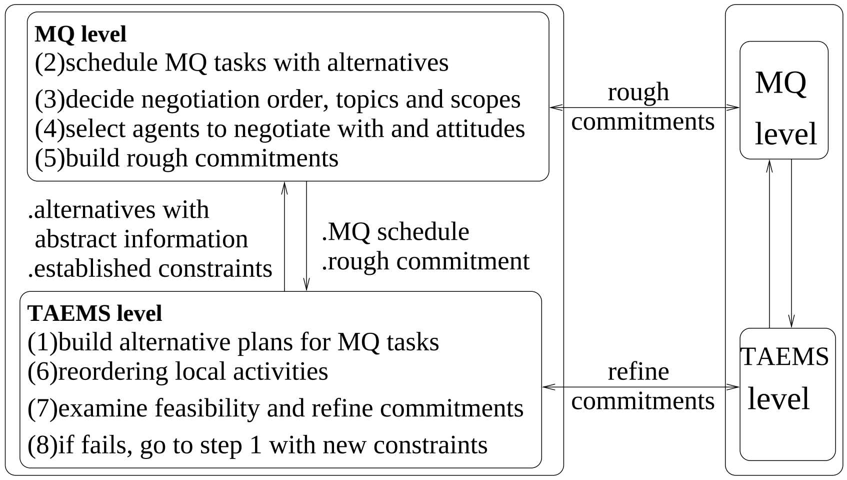

Figure 2: A two-level negotiation framework

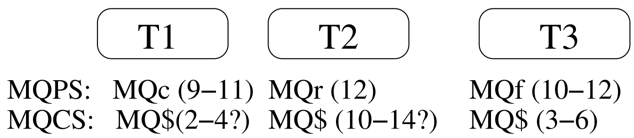

Figure 3: Example of MQ tasks

3 Overview of Basic Ideas

We begin by describing the overall flow of the process as depicted in Figure 2. In Section 4, we will elaborate what is happening in each step. In the two-level framework, an agent has an MQ level view of its local activities, which is a set of potential MQ tasks, each associated with certain $MQPS$ and $MQCS$. Figure 3 shows that agent A has three MQ tasks (the same example as in Figure 1), $T1$, $T2$ and $T3$. $T1$ produces MQ_c from 9 units to 11 units, and it consumes $MQ_\$$ from 2 units to 4 units. The question-mark in the MQ consumption means that there is uncertainty caused by the MQ transferred between agents associated with sub-contracted task, the amount of transferred MQ is not clear at this time point. The amount of the MQ varies depending on what plan is used to accomplish task $T1$. 9 units of MQ_c will be generated using plan P1, and 11 units of MQ_c will be generated using plan P2. The reason is that each plan has different quality (See table 1 and Section 4.1). For each MQ task T, there is a TÆMS task group (task structure) that describes the detailed activities for this task, i.e. the task structure $TG1$ in Figure 4 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 Design-To-Criteria (DTC) scheduler [12], and each plan has different quality, duration and cost characteristics that affect the $MQPS$ and $MQCS$ of the task T (see Section 4.1 for details). This is the first step [step 1] shown in Figure 2, which describes the two-level negotiation framework.

The extended MQ scheduler generates a partial order schedule (Figure 5) that indicates what tasks the agent should attempt to execute, what plans are used to execute these tasks, and the order of the executions. 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. Negotiation on the MQ level is a multi-dimensional negotiation that includes the amount of the transferred MQ, the temporal constraints and the quality constraints of the commitment. When there are multiple related negotiations, the agent needs to decide in what order they should be performed and how [step3]. Also, the agent can select which agents to negotiate with and the appropriate negotiation attitude according to organizational relationships and the negotiation issues [step 4]. The MQ level negotiation builds rough (partially-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]. In this reordering process, the agent optimizes its local schedule by taking advantage of the interrelationships

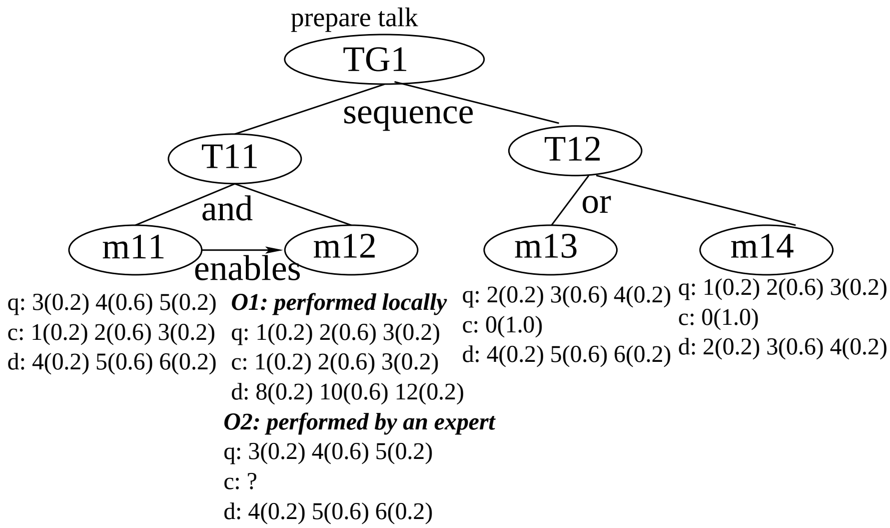

Figure 4: Task structure for T1 *prepare talk*

name	plan	q	c	d	MQPS MQ1	MQCS MQ2
TG1_P1	(m11, m12, m13)	9	4	20	9	4
TG1_P2	(m11, [m12], m13)	11	2	15	11	2

Table 1: Alternative plans for task T1 *prepare talk*

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]. Negotiation on the TÆMS level involves refining those rough commitments as needed. 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 refining any commitments can solve the conflict. Otherwise, if the conflict can't be resolved given all current constraints, the agent needs to discard some commitments (decommits), establish other 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 follow the steps described in Figure 2 to discuss this two-level negotiation in greater detail using the example shown in Figure 1.

4.1 DTC Scheduler Builds Alternatives

The Design-To-Criteria (DTC) scheduler [12] is a domain-independent scheduler that aims to find a feasible schedule that matches the agent's particular criteria request. It is used off-line to build a library of alternative plans for achievement of a TÆMS task group. The three MQ level tasks $T1$, $T2$ and $T3$ are mapped into the task groups $TG1$, $TG2$ and $TG3$ in the TÆMS model. There is a subtask $m12$ of $TG1$ (See Figure 4) that potentially can be contracted to another agent who is an expert on task $m12$. The DTC scheduler works on $TG1$ according to the following different assumptions: $m12$ is executed locally, and $m12$ is contracted to another agent, and generates two alternative plans shown in Table 1. For example, plan TG1_P1 represents the following activities: *find material* (m11), *translate material locally* (m12) and *make slides using powerpoint* (m13). In plan TG1_P2, task $m12$ is performed non-locally, so the cost of m12 is not counted as part of the local plan. Each plan has different performance characteristics, corresponding to an MQ level alternative with different duration, $MQPS$, and $MQCS$. The quality and cost characteristics of a plan affect the $MQPS$ and $MQCS$ of the task, and the influence can be described using domain dependent functions. In this example, the following functions describe how the quality and cost characteristics of a plan P_n are mapped into

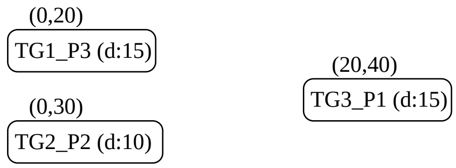

Figure 5: MQ level partial order schedule

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, the resource consumed and produced, and the cost of resource, etc.

This abstraction process can be done off-line, and these alternative plans can be stored in the agent's database. Not all alternatives are used in the MQ level scheduling process. A set of plans is selected according to the 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.

4.2 MQ Level Scheduling

The MQ level scheduler does scheduling for these alternatives of $T1$, $T2$ and $T3$ to find the best schedule S1. If the plan $TG1\_P2$ ($m12$ is contracted out) appears in the scheduler 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 S1 recommends. Suppose the best schedule S1 includes the $TG1\_P2$ plan and two other plans:

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

This is a partial order schedule as shown in Figure 5. Since there is no dependent relationship among these tasks due to MQ resource production and consumption, so they can be executed in any order, as long as the constraints of the earliest start time and the deadline are respected. Agent A compares the utility of the best schedule that includes the contracting plan of $m12$, (S1), with the utility of the best schedule without the contracting plan of $m12$ (S2). S2 is shown as the following:

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

The difference is the utility gained by contracting $m12$ to another agent. It is used by the agents to guide the negotiation on the transferred MQ for contracting $m12$.

Marginal_Utility_Gain($m12$) = Utility(S1) - Utility(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 transferred MQ [16]. The basic constraint of the quality request and the temporal constraint of $m12$ is established based on the TÆMS level schedule ($TG1\_P2$) and the MQ schedule (S1). 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 that 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 we only focus on the negotiation of one non-local task 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 [17, 14, 16], Those approaches all fit into this multi-leveled negotiation framework. In this section, we only focus on 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 a result of the lower-level 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 it in detail. More discussion of other issues in MQ negotiation will be presented in Section 6.

4.3.1 Reward Models

Agents build rough commitments as a result of MQ level negotiation. We use the term "rough commitment" because 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 $[t1, t2]$. $f(c)$ denotes the flexibility of commitment c, $f(c) = \frac{t2-t1-d}{d}$; d denotes the estimated duration of NL. For example, given a commitment c1 on task m12 with time scope [5, 15], suppose the duration of m12 is 5, then the flexibility of this commitment c1 is 1 ($\frac{15-5-5}{5}$). If $f(c) > 0, t2 > t1 + d$, 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 (in terms of when NL can be performed) 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 pre-defined 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. For example, if the commitment c1 on m12 is associated with this pre-paid flexibility model, agent A can request the translator agent to perform m12 during any time range within [5, 15] as long as the scope is no less than the duration 5, such as [5, 11], [7,12], or [9, 15]. The translator agent will accept this request because agent A has already paid for this flexibility.
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 R. β is a parameter that can be adjusted, the agents can negotiate on the value of β. Agent R 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. For example, suppose the commitment c1 on m12 is associated with this dynamic flexibility model, and $v2 = 5$, $\beta = 0.2$, if agent A asks the translator agent to perform m12 during [5,11] and the translator agent can decide to accept this request or not. If it does, agent A will pay $(4 * 0.2 + 1) * 5 = 9$ units of MQ_i to the translator agent for this refinement. Otherwise agent A will still pay 5 units of MQ_i and the translator agent can perform m12 during [5, 15] according to the original commitment

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 Reasoning about Uncertainty

The general approach to accommodate uncertainty in this negotiation framework is described as follows. The uncertainty discussed here refers the uncertainty in the estimation of the execution characteristics (i.e. duration, quality, and cost) of an activity. In the lower-level reasoning process, uncertainties are represented as statistical distributions ($V : \{v_1(p_1); ...; v_i(p_i); ...; v_n(p_n)\}$), which means V has a probability of p_i to have a value of v_i. Uncertainty information is abstracted as:

1. expected value: $E(V)$;

2. marginal value: v_1 and v_n;
3. *measure of above uncertainty*:
   $U(V) = -\sum_i p_i * log(p_i) * \frac{|v_i - E(V)|}{E(V)}$;
4. *probability of above expectation*: $A(V) = \sum_{i|v_i > E(V)} p_i * (v_i - E(V))$.

For example, given $V : \{2(0.4), 4(0.6)\}$:
$E(V) = 2 * 0.4 + 4 * 0.6 = 3.2$
$U(V) = 0.4 * log(0.4) * \frac{|2-3.2|}{3.2} + 0.6 * log(0.6) * \frac{|4-3.2|}{3.2} = 0.093$
$A(V) = 0.6 * (4 - 3.2) = 0.48$
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 measure of above uncertainty $U(V)$ and the probability of the above expectation $A(V)$, the agent chooses the appropriate reward model. If $A(V)$ is large (bigger than a pre-set limit, this pre-set limit can adjusted by the agent based the learning from its experience.) or $U(V)$ is large, the agent chooses the pre-paid flexibility model because of the high probability of future change. Otherwise it chooses the dynamic flexibility model to save some cost on commitment. 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 schedule 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 schedule to see if this change can be guaranteed. If so, an extra cost is added when the change really happens.

4.4 TÆMS Level Negotiation

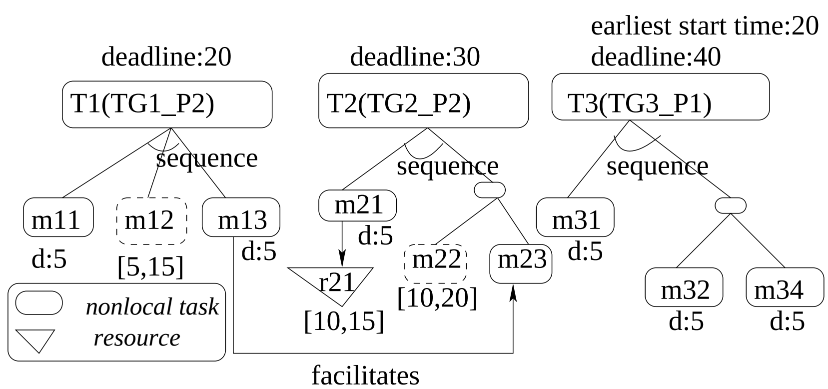

Figure 6: Agent A's TÆMS level tasks

Figure 6 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 $TG1\_P2$, $TG2\_P2$ and $TG3\_P1$. $T1$ has a deadline of 20; $T2$ has a deadline of 30, and $T3$ has an earliest start time of 20 and a deadline of 40. 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 6:

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 7).
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 $m21$.

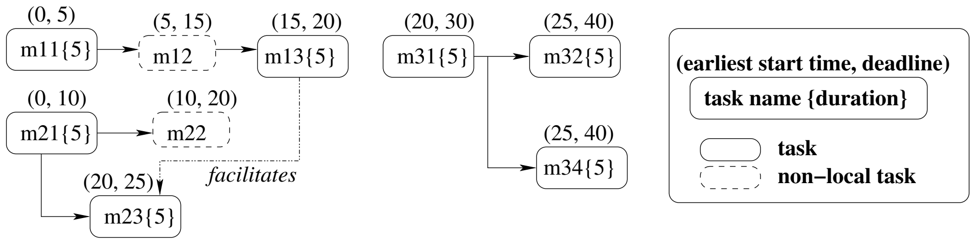

Figure 7: TÆMS level partial order schedule

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. A partial order schedule (Figure 7) is built as a tool for the agent to reason about these temporal constraints. 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 a result of the MQ level negotiation. If this refinement negotiation is successful, agent A can generate a new feasible local schedule (m12 and m22 are executed non-locally):

m11[0-5] *m12[5-15]* m13[15-20]

m21[10-15] *m22[15-20]* m23[20-25]

m31[25-30] m32[30-35] m34[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.

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. For example, if the refinement negotiation fails, agent A may switch to plan TG1_P1 or TG2_P1 to see if it will solve the conflict.
2. Discard some impossible tasks/commitments. For example, agent A can choose to discard task T3 or to negotiate with agent W and ask it perform some parts of task T3 in order to solve the conflict.
3. Reschedule at the MQ level, given the current commitments, tasks, and newly arrived tasks. For example, agent A can reschedule its current MQ tasks including newly arrived tasks, and find what is the best schedule given the current commitments.

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 studies how the two-level negotiation mechanism affects the agent's performance compared to a one-level negotiation, and how the upper-level negotiation (the choice of reward model) affects the lower-level negotiation and hence affects the agent's performance. The setup of the experiment is shown in Figure 8. Three agents were built, agent X, Y and Z. Each agent handles different types of tasks. Type A tasks are handled by agent X, type B tasks are handled by agent Y, and type C tasks are handled by agent Z. Type A tasks are complicated tasks, to accomplish a task of type A, agent X needs to sub-contract task TA2 to agent Y and task TA11 to agent Z. The sub-contracting process of tasks are performed through negotiation as described previously in this paper. New tasks of each type were randomly generated with different rewards and deadlines within certain

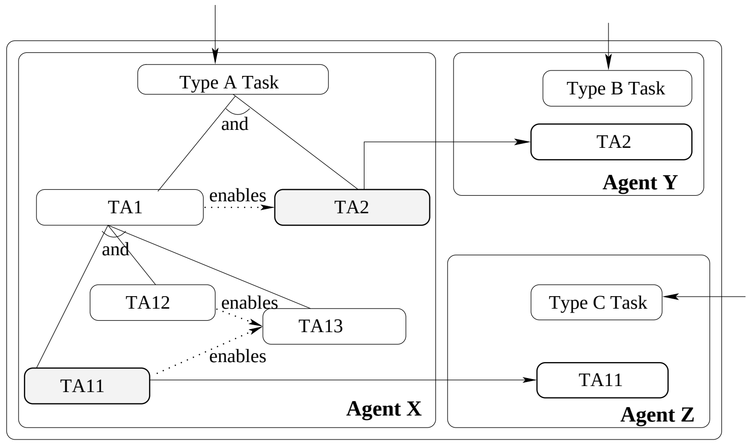

Figure 8: Experiment setup

ranges. Agents decide whether to accept a new task, when to finish it and how to finish it according to the process described in Section 4. Uncertainties are introduced by the execution component that generates the execution time for a task according to its statistical distribution (as the example shown in Figure 4). Some tasks took longer time to be finished than expected, which are referred as "late task". 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[2]. 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. This is a one-level negotiation case.

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 4.3.2.

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 agent X because it is the only agent who needs to sub-contract its subtask to other agents. Figure 9 (each data point is an average over all cases with the same number of late tasks) shows that when 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 a decommitment penalty. The lower-level negotiation helps the agent to adjust its previous commitment with the other agent, so as to avoid missing deadlines of tasks. As the number of late tasks increases, the performance of the dynamic flexibility policy decreases, because the dynamic flexibility policy can not guarantee the success of the lower-level negotiation. Whether the other agent accepts the adjusted request depends 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 number of late tasks. The agent always pre-pays for the flexibility to adjust the rough commitment whether it needs it or not. When the number of late tasks is small (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, 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.

[2]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-level negotiation mechanism is capable of supporting re-negotiation caused by reason of any type.

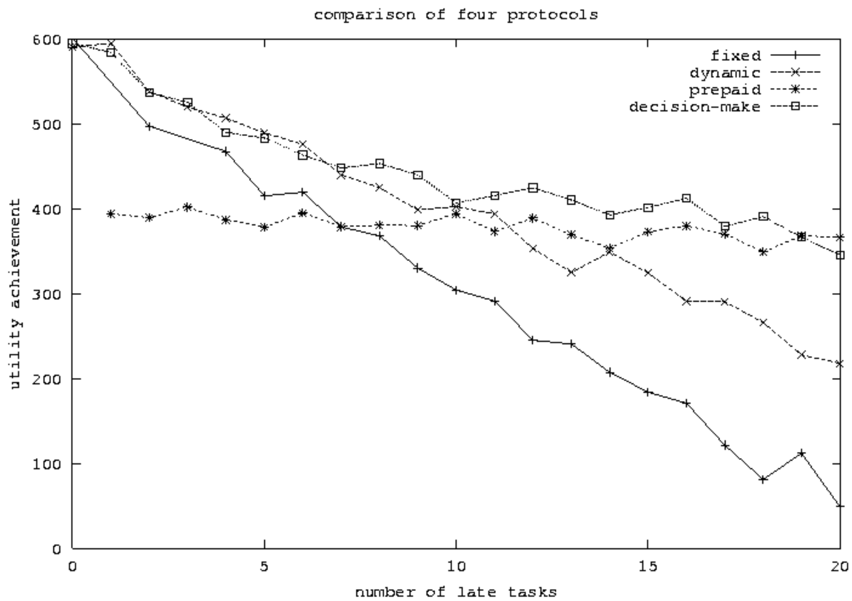

Figure 9: Agent X's performance using different policies when uncertainty changes

6 More Issues in MQ Level Negotiation

In this section, we will address more issues in MQ level negotiation. We will discuss in the framework presented previously, how the agent deal with multiple related negotiations, and how the agent can choose appropriate attitude towards different agents in negotiation.

6.1 Multi-Linked Negotiation

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 issue. Such a situation is call "multi-linked negotiation", where an agent needs to deal with multiple negotiation issues which are related to each other. The relationship among these related negotiation issues are analyzed and classified into two categories: the *directly-linked* relationship and the *indirectly-linked* relationship. In a *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, duration and quality) of issue B directly affect the characteristics of issue A. For example, as shown in Figure 1, the negotiation with a translator agent about the task *translate material* is directly related to the negotiation with the secretary agent about when the college talk should be delivered, because when the translated material is available directly affects when the presentation can be ready. On the other hand, in an *indirectly-linked* relationship, issue A relates to issue B because they compete for use of a common resource. For example, the negotiation about the task *prepare talk* is indirectly related to the negotiation about the task *organize party*, because given the limited computational resource of agent A, when it can perform the task *prepare talk* affects when it can perform the task *organize party*.

In order to minimize the conflicts and maximize the utility in a multi-linked negotiation problem, the agent needs to find an appropriate partial ordering of these negotiation issues and assign suitable scopes for those attributes under negotiation. To solve this multi-linked negotiation problem, we first constructed a partial order scheduler that enables the agent to reason about the time related constraints on different negotiation issues. Two examples of partial oder schedule are shown in Figure 5 and 7. This reasoning tool enables the agent to recognize situations where it is possible to concurrently negotiate individually on different issues without worrying about interactions among their solutions. The partial order scheduler also enables the agent to reason about and manage the flexibility attached to each negotiation issue. Experimental work [14] shows that effective reasoning about

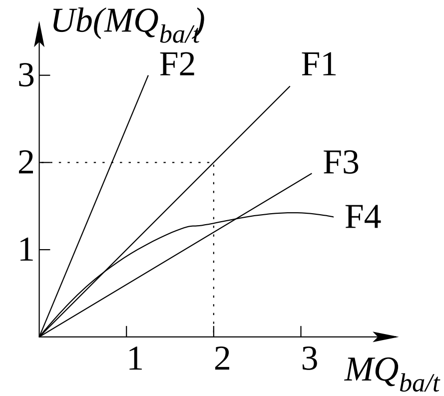

Figure 10: Different mapping functions of $MQ_{ba/t}$

and management of the flexibility in multi-linked negotiation can significantly improve the performance.

Furthermore, we constructed a formalized model for the multi-linked negotiation problem, based on the relationships among negotiation issues. Each negotiation issue is represented as a node with links to other issues representing its directly-linked relationship with those other issues. Each negotiation issue has a set of features (attributes) associated with it. These features either have been decided by other agents or the environmental circumstance or need to be negotiated over. The *success probability* of an issue describes how likely the negotiation over this issue will be successful. It depends not only on the start time of the negotiation, which is decided by the ordering of all the negotiation issues; but also on other features of the issue, such as the reward and the flexibility. Based on this model, a search algorithm can be used to find the best negotiation approach, including how to order the set of negotiations and what values are assigned to those features under the negotiation. A best negotiation approach means that if the agent follows this approach, its expected utility will be maximized. We developed a heuristic search algorithm, which significantly reduces the search effort. Experimental work [15] has evaluated the negotiation approaches generated by this search algorithm and performance is greatly improved over those simple pre-defined approaches.

When the agent performs MQ level negotiation, the mechanisms described above are used to find a near-optimal negotiation approach for multiple related negotiations. Based on this approach, the agent can decide in what order the negotiations should be performed and how. For example, in Figure 1, based on this approach, agent A can decide that in what order the four negotiations should be performed, in parallel, or in sequence (and in what sequence), or some of them in parallel. Also agent A can find how to negotiate on each task so as to avoid conflict with other issues and still have a good chance to get successful results.

6.2 Integrative Negotiation

Given that the other agents in negotiation may have different organizational relationships with this agent, the agent needs to choose appropriate negotiation attitudes toward other agents. In the example shown in Figure 1, when agent A negotiate with the secretary agent, the translator agent, the travel agent, and agent W, it needs to choose different negotiation attitudes depending on the relationship with the other agent and the issue in negotiation.

Traditionally, negotiation research falls into two broad classes: cooperative negotiation [1, 5, 11] and competitive negotiation [9, 10]. In competitive negotiation, agents are said to be *self-interested* and negotiate to maximize their own local utility. In cooperative negotiation, agents work to find a solution that increases their joint utility – the sum of the utilities of all involved agents. Actually, there are many other options between the two extremes of self-interested and cooperative. These other options depend on the importance the agent attaches to the increase of its own utility relative to the importance it attaches to the other agents' utility increases. In a complex agent society, an agent will need to work with other agents from a variety of different organizational positions; hence the agent needs to quantitatively reason about each negotiation session so as to choose an appropriate negotiation attitude. In other words, the choice for the agent is not just "self-interested" or "cooperative", but how cooperative it should be. For instance, in the example shown in Figure 1 agent A may want to be more cooperative to agent W and the secretary agent, than to the translator agent and the travel agent.

This layer negotiation framework supports an integrative negotiation mechanism [17], which allows the agents to choose a negotiation attitude along the spectrum from one that is purely self-interested to one that is completely cooperative. This mechanism is based on the MQ framework described in Section 2. Besides the *goal_related* MQs that are used to represent the progress toward organizational goals, we introduced a new type of MQ - *relational MQ*, as a way of supporting a range of negotiation attitudes.

task name	est	deadline	process time	MQPS
Task 1	10	20	10	[$MQ_{\$}$,3], [$MQ_{hc/t}$, 7]
Task 2	10	30	10	[$MQ_{\$}$,4]
Task 3	10	20	10	[$MQ_{\$}$,9]

Table 2: Three Candidate Tasks

An agent's preference for *relational MQ* represents how cooperative it is with other agents concerning certain negotiation issues. Different preference functions, including both linear functions and non-linear functions, can be used to represent relationships between agents. Let $MQ_{ba/t}$ be the relational MQ transferred from agent A to agent B when agent B performs task *t* for agent A. The amount of the transferred $MQ_{ba/t}$ is used to represented how important this task *t* is for agent A (measured by the increase of its local utility by having this task *t* done). The function that maps into agent B's (virtual) utility reflects how cooperative agent B is toward agent A on this issue *t*.

For example, Figure 10 shows four different functions for mapping $MQ_{ba/t}$ to agent B's utility. Function F1, F2 and F3 are linear functions: $U_a(MQ_{ba/t}) = k * MQ_{ba/t}$.

- When $k = 1$ (F1), $U_b(MQ_{ba/t}) = MQ_{ba/t} = U_a(t)$ ($U_a(t)$ denotes the utility agent A gained by transferring t), then agent B is completely cooperative to agent A.
- When $k > 1$ (F2), $U_b(MQ_{ba/t}) > MQ_{ba/t} = U_a(t)$, then agent B is accommodative to agent A.
- When $k < 1$ (F3), $U_b(MQ_{ba/t}) < MQ_{ba/t} = U_a(t)$, then agent B is partially cooperative with agent A.

The mapping function could also be a nonlinear function (F4) that describes a more complicated attitude of agent B to agent A, i.e., agent B being fully cooperative with agent A until certain amount of $MQ_{ba/t}$ has been accumulated and then becoming self-interested.

This uniform mechanism allows an agent choosing different negotiation attitudes using the same negotiation protocol. By changing the preference function associated with the relational MQ, the negotiation attitude can be easily adjusted according to the agent's organizational goals and the current environmental circumstance. Since the decision making process depends on the output of the MQ scheduler, and the MQ scheduler can handle the *relational MQ* just as other MQs, so this framework can support the agent to choose a negotiation attitude of any type from from purely self-interested to totally cooperative. For example, suppose agent B receives three task proposals with the specifications described in Table 2. Task 1 from agent A with 7 units relation MQ - $MQ_{ba/t}$ and 3 units $MQ_{\$}$ ($MQ_{\$}$ is a normal type MQ). Task 2 and 3 come from other agents with no relational MQ. If agent B is completely cooperative to agent A ($k = 1$), the best MQ schedule produced is as following:
$[10, 20]$ `Task 1` $[20, 30]$ `Task 2`
Agent B will accept both task 1 and task 2. If agent B is self-interested to agent A ($k = 0$), the best MQ schedule produced is as following:
$[10, 20]$ `Task 3` $[20, 30]$ `Task 2`
Agent B will accept both task 2 and task 3, but reject task 1.

7 Summary and Related Work

This paper explores a two-level approach to negotiation in which 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 schedule and accommodate additional constraints and uncertainties of execution behavior. Other researchers have proposed multi-layered agent architectures, but to our knowledge, none has focused on layers of negotiation. One example of a multi-layered agent architecture is the InterRaP [4], which includes three control layers: cooperation layer, plan-based layer and behavior-based layer. This architecture is based on BDI agent model [7], which is different from the utility-driven, quantitatively-reasoning agent control model in our work. Durfee and Montgomery [3] have presented a hierarchical protocol for coordinating multi-agent behaviors. DECAF [4] has also suggested a layered architecture based on separation along functional lines, such as planning, scheduling and executing. However, these architectures have not addressed organizational concerns in the agent's goal selection process, as we do through the MQ framework, and none of them is focused on studying of the layered negotiation as our work does. The two-level scheduling process in this work is related to the early work in hierarchical planning [8] in the sense of decomposing problem into different abstraction level to reduce complexity. However, the contribution of this work is not on planning or scheduling technologies but on the integration of the negotiation decision-making process and the agent's layered scheduling process.

Rather than a stand-alone process, in this work, negotiation 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 layered 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. 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-level 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.

References

[1] S. E. Conry, K. Kuwabara, V. R. Lesser, and R. A. Meyer. Multistage negotiation for distributed constraint satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6), November 1992.

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

[3] E. H. Durfee and T. A. Montgomery. A hierarchical protocol for coordinating multiagent behaviors. In Thomas Dietterich and William Swartout, editors, *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90)*, pages 86–93, Boston, MA, USA, – 1990. AAAI Press.

[4] K. Fischer, J.P. Muller, and M. Pisschel. Unifying control in a layered agent architecture. In *IJCAI95, Agent Theory, Architecture and Language Workshop 95*, 1995.

[5] S. Lander and V Lesser. Understanding the role of negotiation in distributed search among heterogeneous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1993.

[6] David J. Musliner and Kurt D. Krebsbach. Multi-agent mission coordination via negotiation. In *Working Notes of the AAAI Fall Symposium on Negotiation Methods for Autonomous Cooperative Systems*, 2001.

[7] A.S. Rao and M.P. Georgeff. Modelling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann, 1991.

[8] E.D. Sacerdoti. The nonlinear nature of plans. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, 1975.

[9] T. Sandholm and V. Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 126–133, 1996.

[10] Tumas Sandholm and Victor Lesser. Issues in automated negotiation and electronic commerece: Extending the contract net framework. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95)*, 1995.

[11] Sandip Sen and Edmund H. Durfee. A formal study of distributed meeting scheduling. *Group Decision and Negotiation*, 7:265–289, 1998.

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

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

[14] Xiaoqin Zhang and Victor Lesser. Multi-linked negotiation in multi-agent system. In *Proceedings of the First International Joint Conference on Autonomous Agents And MultiAgent Systems (AAMAS 2002)*, pages 1207–1214, January 2002. http://mas.cs.umass.edu/˜xqzhang/pub/ml.pdf.

[15] Xiaoqin Zhang, Victor Lesser, and Sherief Abdallah. Efficient ordering and parameterization of multi-linked negotiation. Technical Report 02-42, UMass, 2002. Research abstract to appear in the Proceedings of the Second International Joint Conference on Autonomous Agents And MultiAgent Systems (AAMAS 2003). http://www.cis.umassd.edu/˜x2zhang/home/pub/02-42.ps.

[16] Xiaoqin Zhang, Victor Lesser, and Rodion Podorozhny. New results on cooperative, multistep negotiation over a multi-dimensional utility function. In *AAAI Fall 2001 Symposium on "Negotiation Methods for Autonomous Cooperative Systems"*, pages 1–10, 2001. http://mas.cs.umass.edu/~xqzhang/pub/NewResultNeg-01.ps.

[17] Xiaoqin Zhang, Victor Lesser, and Tom Wagner. Integrative negotiation in complex organizational agent systems. In *Proceedings of the First International Joint Conference on Autonomous Agents And MultiAgent Systems (AAMAS 2002)*, pages 503–504, July 2002. Extended version available as UMass Tech Report 02-43. http://www.cis.umassd.edu/~x2zhang/home/pub/02-43.ps.