Automatic Formation and Analysis of Multi-Agent Virtual Organization[*]

Qinhe Zheng, Xiaoqin Zhang
Department of Computer and Information Science
University of Massachusetts at Dartmouth
285 Old Westport Rd, North Dartmouth, MA 02747
x2zhang@umassd.edu

March 23, 2005

Abstract

Virtual organization refers to the temporary teaming of enterprises. By sharing physical, human and knowledge resources via information technologies, a virtual organization enables member enterprises to share skills, costs, access to one another's markets and, at the same time decrease the risk of investments. To realize this new generation of business model, the ability to form and operate virtual enterprise is very important. The paper describes our experience gained by implementing a multi-agent system that simulates an artificial marketplace, for which we have derived several decision-making mechanisms in various stages of a virtual organization. We presented a negotiation protocol and a bid selection algorithm for agents to form a virtual organization. We adopted the Motivational Quantities framework to support the agent's local reasoning process. In order to better understand the organizational problem, we adapted a statistical model that predicts the expected rewards of individual agents and the performance of the virtual organization. The comparison and analysis of the results from both the simulation and the model prediction are also presented in this paper.

Keywords: Virtual Organization, Multi-Agent Systems, Agent Control, Motivational Quantities, Simulation.

1 Introduction

A virtual organization can be defined as "*a cooperation of legally independent enterprises, institutions or individuals, which provide a service on the basis of a common understanding of business. The cooperating units mainly contribute their core competences and they act to externals as a single corporation. The corporation refuses an institutionalization e.g., by central offices; instead, the cooperation is managed by using feasible information and communication technologies.*" [4] The virtual organization is a new organization formed by the contributions of resources from several independent enterprises. Of the participating enterprises, a member is designated as initiator agent, who is responsible for task allocation and coordination among the members.

A classical example of a virtual organization is the Agile Infrastructure for Manufacturing Systems (AIMS) project founded by the U.S Government's Advanced Research Project Agency (ARPA). With participating members that include Lockheed, Texas Instruments, and several universities, the goal of AIMS includes the development of mechanisms in both business and technology infrastructures, using national information highways, that would allow companies to very rapidly put together partnerships for the development of complex projects. The set of mechanisms was referred as AIMSNet [12].

This concept of partnership turned out to be what the business world has been looking for. By sharing physical, human and knowledge resources, a virtual organization enables member enterprises to share skills, costs, and access to one another's markets,

[*] This research is supported by UMass, Dartmouth Research Foundation.

Table 1: The Virtual Organization Life Cycle

Identification Phase	Description of product or service to be delivered by the virtual organization, which guides the conceptual design of the virtual organization.
Formation Phase	Rational selection of the individual organizations (partners), which will compose the virtual organization, based in its specific knowledge, skills, resources, costs and availability.
Operation Phase	Control and monitoring of the partners' processes, including resolution of conflicts, and possible virtual organization reconfiguration due to partial failures.
Dissolution	Breaking of the virtual organization, distribution of the obtained profits and storage of relevant information.

at the same time decrease the risk of investments. Through the use of information technologies, the member companies of a virtual organization can work seamlessly across distances, organizations and business boundaries, which enable members to potentially address markets to a global scale. Another benefit for a company to join a virtual organization is the potential of leveraging unused assets. In this case, a company could utilize of otherwise unused resources without interference with its core business.

Generally, the life cycle of a virtual organization can be decomposed in four phases [4], described in Table 1. In order to automate the formation and operation processes, an electronic market infrastructure is needed. Such an infrastructure can be seen as the virtual marketplace where business participants that are geographically distributed can meet each other and cooperate in order to achieve a common business goal. Within this virtual marketplace, individual organizations are the participants and the common business goal is to form a virtual organization that can satisfy a specific business need. To automate the formation and operation process of virtual organizations, agent seems to be an appropriate metaphor and a methodology for system development . A multi-agent system consists of a set of agents that are autonomous or semi-autonomous, which can perform tasks in complex and dynamical environments. There are many similar aspects between multi-agent systems and human organizations: they all consist of intelligent individuals; there are different relationships among these individuals; each individual has only limited knowledge and is resource-bounded; the individuals interact with each other, they coordinate, negotiate, share knowledge, transfer information, and form all kinds of organizations and groups. The traditional development methodologies, such as object-oriented programming, are insufficient in capturing the essences of agents in terms of autonomy and pro-active nature. The recent studies of agent-based engineering have enhanced the development methodology for multi-agent system, and made it easier to implement virtual organizations. Therefore, it is natural to view virtual organizations as multi-agent systems. The virtual building company described in this paper is a multi-agent system where each individual agent exhibits capabilities to be autonomous, has the ability to interact with other agents in virtual market and make rational decisions in changing environments.

The objectives of this research include: to implement a multi-agent system that supports the simulation of artificial marketplaces, to derive mechanisms for decision-making in various stages of a virtual organization (i.e. as described in Table 1), and to perform experiments to evaluate and verify those mechanisms in order to better understand the organizational problems. Our work is mainly focused on the decision-making process of the member agents (each organization is represented by a software agent) during the various stages of a virtual organization, such as the partner selection process during the formation phase and the task selection process of individual agent during the operation of a virtual organization. We use Motivational Quantities (MQ) framework to support the agent's local decision-making process. Additionally, we adapt a statistical model to predict the task conflicts and expected reward for each agent during the lifetime of a virtual organization.

The remaining of paper is organized in the following way. In Section 2, we first introduce a virtual building organization scenario that is used as an example through this paper . We then describe the detailed process in the various phases of the VO's life-cycle in Section 3: a negotiation protocol for the formation phase, a recursive best-first search algorithm (RBFS) for the partner selection process, the agent's utility mapping function based on the MQ framework, the organization's penalty policy for lack of commitment. We will also present an analytical model to predict the behavior of the agents and the organization in Section 4. Then we will describe our experimental work and results in Section 5. Related work will in discussed in Section 6. Finally we will present our conclusions and the directions for future work in Section 7.

2 Scenario: Virtual Building Organization

One of our main objectives is to implement a multi-agent system that supports the simulation of an artificial marketplace; in that aspect we have developed a scenario as the base of our model. A real estate developer, named Concrete Developer, has recently won the right to develop a large suburban area for residential use. Concrete Developer has always relied on a single outside contractor, who in turn enlists a group of sub-contractors, to construct the residential buildings. However, after a careful analysis, it decided that it would be much more profitable and effective to form a virtual organization. The developer partitions the building process into 5 partial processes, namely framing, foundation, electrical work, plumbing, and finishing, assuming they must be completed in sequence. The developer makes the initial proposal of forming a Virtual Building Company to the sub-contractors in its marketplace. The individual enterprises can then bid for these partial processes. After the developer has received substantial bids from the individual contractors, it then selects a group of bids that meets its highest expectation based on multiple criteria, such as competence, availability, etc. Once the virtual organization is formed, it goes into the operational phase. During the operational phase, a buyer may request for a house at any given time (the negotiation process between the buyer and the developer is omitted here for simplicity). After receiving a buyer request, the developer notifies individual participants of the virtual organization, who may or may not commit to a subtask based on their own decision-making mechanisms. The developer accepts a buyer's request only when it has all the commitments necessary to complete the whole construction task. Only when all the subtasks are completed, the developer can collect money from the buyer. An agent may receive service request from the developer agent, it may also receive service request directly from the buyers or another virtual organization that the agent also belongs to.

There are three type of agents in our model, the initiator agent (the developer), individual ”worker” agent (an agent that is capable of partial process), and the buyer agent. The initiator agent is the one who takes the initiative in the formation of a virtual organization and is responsible for task allocation and management during the operation. An individual agent is a self-sustained entity, it may receive service request directly from the buyer and is free to join any virtual organization. The buyer agent is the simplest one; its sole purpose is to send service request to any virtual organization or any individual agent. Each of the three types of agents can be instantiated into any number of distinct agents by specifying its characteristics such as name, competencies, availability and etc.

We have successfully implemented the system that models the virtual building company. It is implemented using Java Agent Framework [8] and runs under the MASS simulator [9]. While this system models the work-flows of a construction company, the same procedures could be used for any virtual organization. When implementing the system, we have ignored the business rules that are associated with the specific industry, but mainly focused on areas where agents need to make rational decisions. Discussion of such decision-making processes is summarized in the following sections.

3 The Formation and Operation Process

In this section, we describe the details in the VO's formation and operation process, including how to form a VO and how individual agents make decisions, and how those decisions affect the VO.

3.1 Negotiation Protocol

When the initiator is planning to make the initial proposal, it needs a set of evaluating criteria to select the most favorable group of members (each member has a set of attributes, such as the cost and the quality of task performance, and its availability, etc.). It realizes that the competence of the resulting virtual organization is related to the constraints attached with each member's attributes. For example, if an agent can perform only n unit of a partial processes, then the maximum achievable VO output couldn't exceed n units. Therefore, the initiator must determine the set of evaluation criteria and impose a preference order on the attribute values. On the other hand, individual agents are faced with the decision of whether or not to join a particular virtual organization. Joining a virtual organization is beneficial but it is not always cost free; for instance, in a practical situation, a member agent may be required to adopt standard business process and information technologies in order to facilitate communication and transaction process in these virtual organizations. Therefore, it must carry out a cost-benefit analysis, based on the information provided by the initiator agent and the degree of belief it has in the initiator agent, before joining a virtual organization.

A negotiation protocol is needed for both the initiator and the individual agents in support of their decision-making during the formation phase. In our model, the proposal sent by the initiator agent includes the following information: the type of task (building construction) needed for the organization, the estimated work load for each type of subtask (partial processes), and the estimated profit of the organization. The bid from the potential participant includes the following information: the type of partial process the agent is capable of, the number of units it will contribute to the organization (capability), and the profit sharing rate (how much it requests from the virtual organization's profit). In our simple model, this protocol is sufficient for individual agent to decide whether or not to join a virtual organization and for an initiator to evaluate a bid.

3.2 Partner Selection Process

Before the proposal can be made, the initiator needs to decompose the whole product/service process into several partial processes. Usually this can be done by human being assisted by a suitable modeling techniques, here we assume that it has already been done. Once the initiator agent has identified the partial processes, the partial processes will be distributed to the agents so that each agent can make its own contribution depending on its specialty, hence an allocation process is needed. In this marketplace, the individual agents complement each another in their service offer, i.e., different enterprises cover different partial processes, which resembles a horizontal allocation of a real world business practice. To find the members, we could first search for an agent that will deliver a partial process, then continue looking for other agents that are able to deliver those complementary services, and so on. The search is finished when the organization is self-contained and does not require any more services from other service provider in the marketplace.

However, the objective of the selection process is not only just to select a group of members that would form a self-contained virtual organization, but also to form a virtual organization that would maximize the profit. There are two different approaches in terms of maximizing the profit given the set of partial processes, and the estimated work load L. The initiator may try to maximize the profit of the virtual organization. Given the profit is proportionate to the work load, so a maximized work load L will generate maximized profit $R * L$. In this case, the initiator will prefer agents with higher capabilities. However, being a self-interested agent, the initiator could be more interested in maximizing its own profit. The initiator's profit depends on the profit

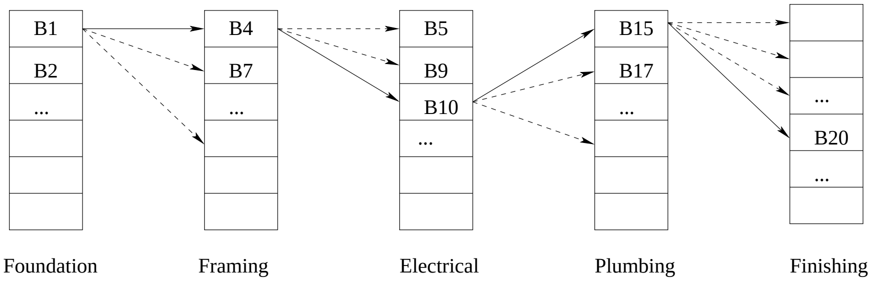

Figure 1: Partner Selections (Note that this graph does not show all the combinations. In fact, each bid in a bin can be linked to any bid in the next bin. Solid lines represent the final selection and dash lines show other possibilities.)

of the organization and the profit to be handed out to other agents in this organization. Assume that a bid B_i contains the following information: type of task T_{bi}, number of commitments promised L_{bi}, profit sharing rate S_{bi}, then the profit for the initiator would be: $R \cdot L' \cdot (1 - Sum(S_{bi}))$ where L' is the practical work load ($L' = min(L_{bi})$, and $L' <= L$).

We adopt the second approach in partner selection process for our virtual building company. The core of this procedure is a recursive best-first search algorithm (RBFS) with some heuristics. It works in the following way. First, it groups the bids into different bins according to which task they are bidding on (Figure 1). Then it selects one bid from the first bin which will maximize the initiator's profit, but it also remembers the second best choice. It goes to the next bin and finds the best bid that can be combined with the choice from the previous bin: if the initiator's expected profit from this set of combined bids is no less than the second best choice of the previous bin, it continues to the next bin; otherwise, it will unwind to the previous bin and choose the second best and proceeds from there. It continues this process until an optimal solution is found.

In our model, the selection criteria includes only the agent's capability (number of commitments promised) and the profit sharing rate it asks. It is obvious that these two criteria are inadequate in real world application. In a more realistic application, we shall also consider other attributes of a bidder, such as the quality of its contribution and the time required to complete a partial process. Each additional variable will inevitably increase the complexity of the selection process, thus, this process could easily become intractable. One way to get around this is to use a set of screening filters to eliminate undesired bids from the selection pool before a search algorithm is applied to find an acceptable solution.

3.3 Penalties for Lack-of-commitment

An additional problem in virtual enterprise is that the agents participate in the marketplace are self-interested, trying to maximize their local utilities. This implies that, whenever it is beneficial, agents may lie during the selection process in order to be more attractive to the initiator agent. Therefore, an enterprise may need an incentive to encourage the agents to maximize the profit of the enterprise. During the bidding process, an agent is required to specify the number of the partial processes it is capable/willing to perform to a virtual organization. However, during operation phase the agent may, in self-interested fashion, favor other opportunities and leave the commitment to the virtual organization unfulfilled. Unless there is an mutual trust established among the member companies, especially between the initiator agent and the individual agents, a penalty for less commitments than what it has promised is the most straightforward incentive and is also easy to be implemented. Depending on how the penalty is calculated, there are different penalty policies. A linear penalty policy has a fixed penalty rate for each unfulfilled commitment. A progress-based penalty policy has a decreasing penalty rate as more commitments have been fulfilled. It charges a heavy penalty if the agent can not fulfill a minimum percentage of its promise, and it charges a much less penalty if the agent has fulfilled a certain percentage of obligation. For instance, a progress-based penalty policy can be stated as the following: if the agent can not

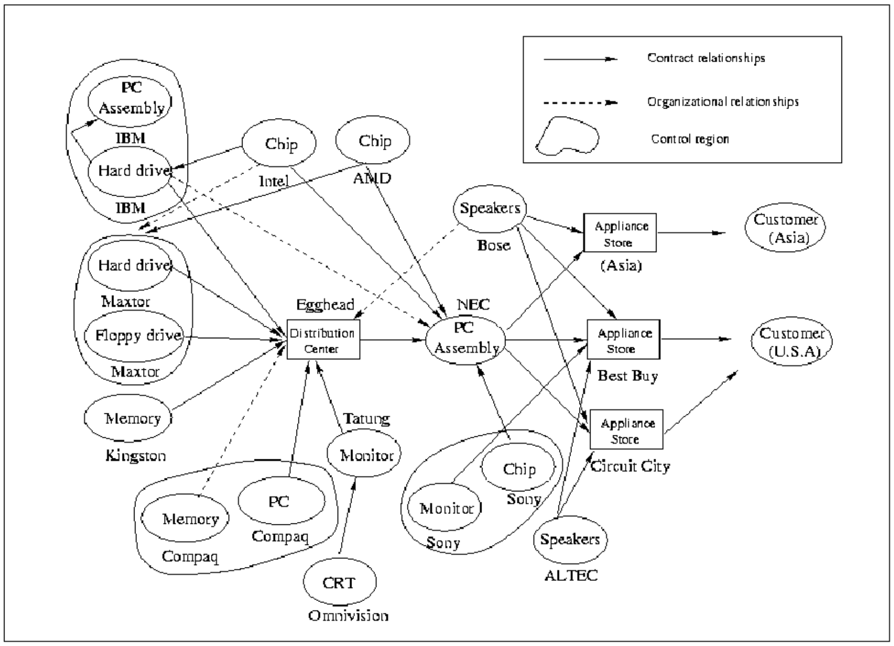

Figure 2: Complex Agent Organization Relationships

fulfill 30% of its promised commitments, there is 100 units penalty for each unfulfilled commitment; if the agent has fulfilled 90% of its promise, there is only 10 units penalty for each unfulfilled commitment.

To react relationally toward the penalty of lack-of-commitment, the agent needs to incorporate the penalty policy into its local decision-making process. In our model, this is implemented by introducing a control parameter in the utility mapping function (See Section 3.5 for more detail) associated with the organization task. By adjusting this parameter, different penalty policies can be reflected in the agent's decision-making process, so the agent can balance the profit and penalty when making selection on different types of tasks.

3.4 Motivational Quantities

For an organizationally situated agent, it must interact with agents within and out of its organization; therefore, it is essential that agents must model their organizational relationships and reason about the value of utility of interacting and coordinating with particular agents over particular actions [12]. In a more complicated situation, an organizational agent may belong to multiple virtual organizations with different, or even conflicting, goals and objectives, and the cooperation attitude between a local agent and others may range from fully self-interested to full cooperative. To further complicate the matters, real agents are hindered by bounded rationality, limited resources, and imperfect knowledge of the environment. The interactions among the agents may result in a complex relationship diagram similar to Figure 2. The dash line represents virtual organizational relationship established among agents who have long-term business relationship. When examining a scenario like this, there is the need to model the different motivational factors that influence agent decision-making, such ability is a requisite for the agents to act rationally given their organizational context.

In a complex organizational context, an agent may receive service request from different agents in order to make progress toward different goals. If the agent cannot perform all the tasks, it has to select a subset of the tasks to perform and determine an appropriate sequence to perform them. This problem faced by an agent can be categorized as a real-time action-selection-

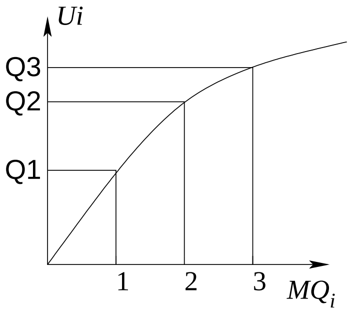

Figure 3: Motivational Quantities and Utilities

sequencing problem where an agent has n candidate tasks and alternative ways to perform the tasks. Tasks have deadlines and other constraints as well as different performance properties, e.g., consuming different resources or producing results of varying quality. The appropriate decision on how to perform them depends on the agent's context, which includes its relationships with other agents, shared organizational goals and individual goals, commitments made to other agent, and resource limitations. It is in this context that [15, 16] suggests to quantify motivational factors using Motivational Quantities (MQ) Framework.

The MQ framework is an agent control framework that provides the agent with the ability to reason about which tasks should be performed, when and how to perform them. The reasoning is based on the agent's organizational concerns. The basic assumption is that the agent has multiple goals related to the multiple roles it plays in the agent society. The progress toward one goal cannot substitute for the progress toward another goal. Motivational Quantities (MQs) are used to represent the progress toward organizational goals quantitatively. MQs are consumed and produced by performing MQ tasks. Each agent has a set of MQs which it is interested in and wants to accumulate. Each MQ_i in this set represents the progress toward one of the agent's organizational goals. Each MQ_i is associated with a preference function (utility curve), U_{f_i}, which describes the agent's preference for a particular quantity of the MQ_i. The MQ framework thus provides an approach to compare the agent's different motivational factors through a multi-attribute function. Not all agents have the same MQ set. Different agents may have different preferences for the same MQ.

The agent's overall goal is to select tasks to perform in order to maximize its local utility through collecting different MQs. MQ tasks are abstractions of the primitive actions that an agent may perform. The agent compares and selects tasks that are associated with different organizational goals. Each MQ task T_i has the following characteristics: earliest start time (est), deadline (dl_i), and process time needed to accomplish the task (d_i).

The MQ scheduler schedules current potential MQ tasks, and produces a schedule for a set of MQ tasks, specifying their start times and finish times. The scheduler takes the following factors into consideration: the MQ consumed and produced by performing task T_i, duration d_i, the earliest start time est_i and the deadline dl_i of each MQ task, and the agent's current accumulation of MQs. Notice that MQ is always being evaluated in the context of agent's current MQ accumulation state. For example, Figure 3 shows a single utility curve for a single MQ_i. The first one unit MQ_i brings the agent Q_1 units of utility U_i . After the agent has collected 2 units of MQ_i, the additional one unit of MQ_i brings the agent additional $Q_3 - Q_2$ units of utility U_i. $Q_3 - Q_2$ is not necessarily equal to Q_1, they are all calculated based on the utility curve associated with MQ_i.

The MQ framework provides the comparison of tasks that need to be performed for different reasons: for different organizational goals, for other agents to gain some financial benefit or favors in return, for cooperation with other agents, etc. It also supports different utility functions that relate the execution of tasks to the importance of organizational goals. In summary, the motivational qualities framework provides an agent with the capability to reason about different goals in an open, dynamic and

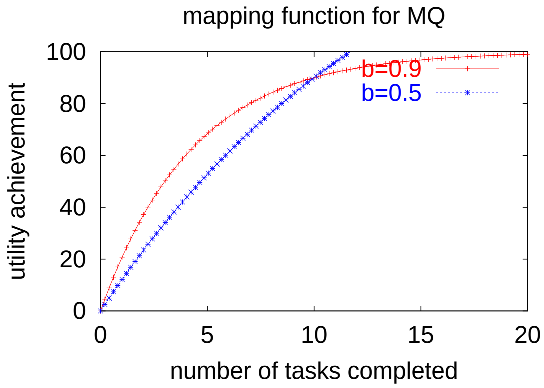

Figure 4: Mapping Functions in Motivational Quantities

large-scale multi-agent system.

3.5 Utility Mapping Function of MQ

In a virtual organization, each member agent receives service requests not only from this organization (referred as *organization task*), but also from other organizations (if the agent belongs to multiple organizations) or directly from customers (*outside task*). When there is conflict between different tasks, the agent needs to decide which task to commit. The MQ framework provides such a mechanism for keeping the different motivational concerns separate, because they represent progress that are not interchangeable. For example, the completion of task a is a progress toward objective A, but does not necessarily present a progress toward objective B. MQ enables agents to compare different types of tasks, the costs and the benefits of a particular courses of actions.

Based on MQ framework, we assume that each different type of task produces a different type of MQ. For instance, tasks from organization A produce $MQ_{organizationA}$, tasks from organization B produce $MQ_{organizationB}$, and tasks from direct customers produce MQ_{direct}. There is a utility mapping function associated with each type of MQ, and it reflects how the agent evaluates this task in terms of the contribution to its local goals and objectives. To focus on the study of virtual organization, we assume that the outside tasks only produce monetary value, the mapping function for MQ_{direct} is expressed as $f(x) = y$, which maps each unit of monetary value into one unit of local utility.

It is more complicate to evaluate an organization task, the agent needs to consider a number of issues: how important the organization's achievement is to this agent, how many commitments it has made for this organization, what the penalty policy is, etc. Here we propose a mapping function that takes into the consideration of the number of commitments the agent has promised to a virtual organization, the maximum expected reward from/of a virtual organization and the penalty policy of the organization. The utility gain from performing an organization task can not simply be measured by monetary value, since the agent not only gets money in return, but also gets benefits from having a good relationship with the initiator agent. The mapping function is expressed as $f(x) = a \cdot \frac{1}{b} \cdot (1 - ((1-b)^{\frac{1}{c}})^x)$, where a is the expected utility from/of the virtual organization depending on how important the agent feels about the organization's achievement. For example, if the expected utility of the organization is 1000, and the profit sharing rate for this agent is 20%. a can be set as 1000, which means the agent views the organization's achievement as its own; a can be set as 200 (1000*20%) if the agent only cares about its own gain; and a also can be set as any other number between 200 and 1000, depending on how much emphasis the agent has on the organization's achievement. c represents the number of commitments the agent has made towards this organization. b is a control parameter and $0 < b < 1$, which works with c together to reflect the penalty policy (See details in the following example). The function is a derivation of a general function $a * (1 + b^{-x})$

which produces an upward decreasing curve. By adding a third variable c to the formula, the agent would have more control on how to fulfill its promise to the organization. The intention is the agent would try to fulfill its promise to the organization; afterward the utility gain from performing organization tasks would slow down. The mapping functions are illustrated in Figure 4. In the first case, we have a=90, b=0.9 and c=10. As we can see from the graph, it has an upward decreasing curve; the agent has the tendency of preferring organization task rather than the outside task, which can be visualized as having a linear curve. This is especially true in the beginning, and eventually the utility gain became flat as the first few commitments have been fulfilled. This actually reflects the progress-based penalty policy, where the penalty rate decreases significantly after the agent has fulfilled a certain percentage of obligation. Depending on the organization's penalty policy, one can modify the value of parameter b in order to adjust the agent's attitude toward the organization task. For example, if the organization has a linear penalty policy, the first mapping function may not be a good idea since it only favors the first few commitments. By changing the b value from 0.9 to 0.5 (represented by an near linear curve, assuming to be steeper than linear curve of the outside task), the agent would have the preference on the organization task before all commitments has been fulfilled.

4 Analysis Based on a Statistical Model

In order to analyze the agent's behavior and understand how it affects the organization, we adapted a statistical model that was originally presented in [13]. We modified this model so it can be used in our VO scenario.

4.1 The original model

The expected reward for an organizationally situated agent is straight forward if the agent is expecting one type of task without scheduling conflicts. However, this is rarely the case, since an agent may coordinate with multiple agents and/or belong to multiple organizations. Consequently, an agent may receive multiple service requests at any given time. Being self-interested by nature, an agent must make decision on which tasks to perform and in what order to perform them. [13] proposed a generic statistical model that anticipates the probabilities of conflict between any two types of tasks for a local agent and the expected reward for the agent[1]. This generic model assumes a simple agent organization, which consists of three agents, A_1, which is the initiator agent, has task T_1 coming in, of which there are two subtasks $sub2$ and $sub3$ that need to be sub-contracted to A_2 and A_3 respectively. At the same time, T_i arrives at A_i with a probability of $\frac{1}{r_i}$ at each time unit. For each task there is a number of parameters associated with it, as shown in Figure 5 (which also illustrates their relationships). For task T_i, e_i, dur_i and sl_i are uniformly distributed within the ranges $\left(a_i^e, b_i^e\right]$, $\left(a_i^d, b_i^d\right]$, $\left(a_i^s, b_i^s\right]$; and for task sub2 and sub3, e_i, dur_i and sl_i are uniformly distributed within the ranges of $\left(a_{1i}^e, b_{1i}^e\right]$, $\left(a_{1i}^d, b_{1i}^d\right]$, $\left(a_{1i}^s, b_{1i}^s\right]$.

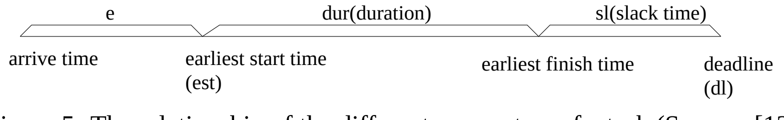

Figure 5: The relationship of the different parameters of a task (Source: [13])

An agent needs to choose which task to execute when and only when there is a conflict between tasks. A task of type i is in conflict with a task of type j (whether it comes before task i or after) if and only if the following two inequalities are both true:

$$dli - estj <= duri + durj,$$

$$dlj - esti <= duri + durj$$

[1] The material presented in this section is quoted from [13], the purpose is to give readers a brief overview of this model in order to understand our following work. Each equation can be solved using the basic statistical parameters. Details are omitted here and can be found in [13].

By rewriting the two inequalities in term of est, dur and sl, we get: $sl_i - dur_j \leq est_j - est_i \leq dur_i - sl_j$
To calculate the expected reward for an agent at any given time, we need first calculate the probability that an arriving task type i is in conflict with a task of type j.

$Pcij = P(sli - durj <= estj - esti <= duri - slj)$

Given the probability of conflict, we can calculate the expected reward for each agent. For A_2 and A_3, there may be two types of tasks coming in at any moment: the direct task T_i (can be viewed as outside task) and the subcontract task sub_i (can be viewed as organization task) with a probability of $1/r_i$ respectively, where $i = 2, 3$. Let us look at them one by one.

When a direct task T_i for A_i arrives, it accumulates reward only under one of the following circumstances:

1. There is a conflict between T_i and a subcontract task sub_i and there is not conflict with other direct tasks. In addition, the direct task reward is greater than the utility of the subcontract task that it is in conflict with, i.e., $R_i > Rn_i$. The expected reward gained by executing the new task in this case is:

$$ER_i^{(1)} = Pc_{1i,i} \cdot (1 - Pc_{ii}) \cdot E(R_i | R_i > Rn_i) \tag{1}$$

2. The only conflict caused by this task is with another direct task T_i'. In addition, the new reward is higher than that of T_i'. The expected reward gained by executing this task under this condition is:

$$ER_i^{(2)} = (1 - Pc_{1i,i}) \cdot Pc_{ii} \cdot [E(R_i | R_i > R_i') + \frac{1}{2} E(R_i | R_i = R_i')] \tag{2}$$

3. There are conflicts with both another direct task and a subcontract task. In addition, the reward gained by the new direct task is the highest.

$$ER_i^{(3)} = Pc_{1i,i} \cdot Pc_{ii} \cdot [E(R_i | R_i > Rn_i \& R_i > R_i') + \frac{1}{2} E(R_i | R_i > R_{ni} \& R_i = R_i')] \tag{3}$$

4. There is no conflict caused by the new task.

$$ER_i^{(4)} = (1 - Pc_{1i,i})(1 - Pc_{ii}) \cdot \frac{ar_i + br_i}{2} \tag{4}$$

Similarly, when a subtask sub_i arrives at A_i, A_i will choose to commit to it under certain conditions, but it can accumulate this reward only when the other agent decides to commit to the other subtask as well. Therefore the expected reward will be:

$$ER_i^{(5)} = Pcommit_2 \cdot Pcommit_3 \cdot R_{1i} \tag{5}$$

where $Pcommit_i$ is the probability of agent A_i commits to the subtask sub_i ($i = 2, 3$).

Now we have the expected reward that A_2 or A_3 collects at each time unit:

$$ER_i = \frac{1}{r_i}(ER_i^{(1)} + ER_i^{(2)} + ER_i^{(3)} + ER_i^{(4)}) + \frac{1}{r_1} ER_i^{(5)} \tag{6}$$

Although this model does not explicitly express the expected reward for virtual organization, we could derive the reward by first calculating the probability P of commitment from all agents, then the expected reward for the virtual organization ER_v at any given moment would be: $ER_v = \frac{1}{r_1} \cdot P \cdot Rn_i$.

```
Algorithm 4.1 function createLookupTable (organizationTask, outsideTask) returns a lookup table
    table ← MAKE-TABLE(empty)
    arrival_number ← expected number of organization tasks
    current_accepted ← 0
    conflict_with_outside_task ← CONFLICT(organizationTask,outsideTask)
    conflict_with_itself ← CONFLICT(organizationTask, organizationTask)
    loop for i from 1 to arrival_number do
        reward ← UTILITY_GAIN(organizationTask,current_accepted)
        probability_accept ← (1 - conflict_with_itself) · (1 - conflict_with_outside)
            + 0.5 · (1 - conflict_with_outsideTask) · conflict_with_itself
            + 0.5 · conflict_with_outsideTask · conflict_with_itself · $P_{reward\_greater\_than\_outsideTask}$
            + conflict_with_outside · (1 - conflict_with_itself) · $P_{reward\_greater\_than\_outsideTask}$
        table ← MAKE _ ROW(i,current_accepted,probability_accept,reward)
        i ← i + 1
        current_accepted ← current_accepted + probability_accept
    return table
```

Figure 6: Lookup Table Algorithms

4.2 The modification: introduce a look-up table algorithm to handle the dynamic mapping of MQ

In order to model the Virtual Building Company, we need modify the statistical framework described above to reflect the difference between our virtual organization and the generic multi-agent system described in the original model. In the original model, it is assumed that the reward of a task is uniformly distributed within the range $(a_i^r, b_i^r]$. In our framework, it is only true for outside tasks. For organization task, the agent measures its reward using a utility mapping function based on the MQ produced by performing this task. Because MQ is always being evaluated in the context of agent's current MQ accumulation state, so the reward of an organization task also depends on the agent's current contribution toward the virtual organization. The first organization task with 1 unit $MQ_{organization}$ and the second organization task with 1 unit $MQ_{organization}$ may produce different amount of local utility for the agent, depending on how much $MQ_{organization}$ the agent has collected.

To solve this problem, we create a lookup table, which calculates the expected reward for an organization task at a given time. This calculation is based on the estimation of the agent's current MQ accumulation state, which is based on the estimation of how many organization tasks have been accepted previously. To estimate how many organization tasks have been accepted, we need calculate the probability of conflict and compare the rewards of different tasks. This algorithm would create a table similar to Table 2. Once we have this lookup table, we can have a function $R(t)$, which calculates the expected reward for that given time when provided with the current time unit t. As in the original model, the assumption is that we have prior knowledge of all agents, including the frequency of the arriving organization task and the duration of the virtual organization. To find the expected reward at a given time, we calculate the estimated arrival number using $arrival\_number = (duration+1)/\mathrm{r_j}+1$, and use the reference lookup table to find the estimated accumulation of MQs (current_accepted) and the corresponding utility gain (expected reward at time t). The overall expected reward for the agent during the operation of a virtual organization can be calculated using formula 6, by replacing Rn_i the function $R(t)$.

Based on the new formula derived we have implemented the statistical model as a stand-alone application, thus the comparisons between the simulation runs and the model predictions can be made.

5 Experiments

The experiments were designed to verify the correctness of a set of mechanisms we developed with a goal of unveil any relevant information base on the data we gather from both the simulation and the statistical prediction. Furthermore, we would like to study the agent's behavior under different control settings. By alternating the parameters of different types of tasks for an agent, we would like to see the effect of the mapping function on agent's promise to the organization, the agent's local utility, and the organization's utility.

5.1 Verification of Agent Model

The first set of experiments is to verify the statistical model through the simulation results. In an artificial marketplace, individual agents exist in the virtual environment and the market demands are generated by the buyers. By controlling the parameters/characteristics of agents and the frequency of market demands, we can perform analysis on the outcome of the artificial marketplace. The statistical model takes parameters $(freq, \left(a_i^d, b_i^d\right], (a_i^s, b_i^s], (a_i^r, b_i^r])$for each task type, which are corresponding to its frequency, the range of deadline, the range of the slack time, and the range of reward by performing such a task. These parameters enable each agent to calculate the probabilities of conflict between any two tasks and the expected reward. By changing the corresponding parameters for each agent in Virtual Building Company, we are expecting to have the matching results, or at least statistically close to the model prediction.

The experiment setting consists of five agents in a virtual organization with a running time of 3000 clock cycles. Each agent uses the following parameters: $(40, (25, 30], (0, 10], (750, 1500])$ for an outside task, and $(35, (20, 30], (0, 10], a \cdot \frac{1}{b} \cdot (1 - ((1 - b)^{\frac{1}{c}})^x))$ for an organization task, where the reward/utility of an organization task is calculated using the utility mapping function, with a = 30000, b = 0.9, and c = 20. The mapping function is used to calculate the utility of an organization task, so it can be compared to the utility of an outside task, which is direct mapping from the monetary value. Given this set of parameters, we are expecting to have, on average, 75 occurrences of outside task and 85 occurrences of organization task. In order to calculate the conflicts in the agent model, we keep a stack of all the arriving tasks during the VO operation, regardless whether a task is being executed or not. At the end of the system run, we count the number of conflicts for a particular task type and divide it by the total number of occurrences.

The probability of conflict from both the statistical model and the agent model are summarized in Table 2. A similar experiment was conducted with a different set of parameters: $(80, (20, 30], (0, 10], (750, 1500])$ for an outside task, and $(25, (20, 30], (0, 10], 1000, a \cdot \frac{1}{b} \cdot (1 - ((1 - b)^{\frac{1}{c}})^x))$ for an organization task, where a = 30000, b = 0.5, and c = 20; with expected 38 outside tasks and 120 organization tasks. The results are also summarized in Table 2. We found that the probability of conflict in simulation is well predicted by the statistical model, though the prediction is a little bit higher than the simulation result. This may be explained by the fact that the simulator uses a scheduler that schedules all tasks fall into a fixed time window, hence the conflict between tasks that belong to different scheduling windows are not caught by the simulator.

5.2 Effects of MQ Mapping Function

Our second set of experiment was to investigate the effect of the mapping function on an agent's promise to the organization, its local utility, and also the overall virtual organization's performance. We used $U(x) = a \cdot \frac{1}{b}(1 - ((1 - b)^{\frac{1}{c}})^x)$ as the mapping function to evaluate the organization task. As described earlier, this utility function has an upward decreasing curve with the

Table 2: Comparison of Conflict Probability (P(Out, Out) refers the conflict probability between outside tasks, P(Out, Org) refers the conflict probability between an outside task and an organization task. Similar explanations are for P(Org, Out) and P(Org, Org).

	Experiment 1		Experiment 2	
	Prediction	Actual	Prediction	Actual
P(Out, Out)	0.26	0.21	0.57	0.6
P(Out, Org)	0.81	0.72	0.62	0.46
P(Org, Out)	0.82	0.74	0.68	0.44
P(Org, Org)	0.25	0.18	0.54	0.45

ability to change shape as desired. Such a mapping function is useful especially in modeling the lack-of-commitment penalty and the intangible gains from the organization task. Within the context of our agent model, the function is useful only when the MQ scheduler is called during the operation phrase, in order to select tasks from two or more conflicting tasks.

There were several problems we encountered during this experiment. It turned out that our first experimental setup produced tasks that potentially required the use of MQ scheduler (often an outside and an organization task at the same time); however, there was a high probability that the arriving tasks are in conflict with previously committed tasks. As a result, one of the arriving tasks (or both) was rejected automatically. Therefore, the MQ scheduler was never called and so the mapping function did not come into play. Another problem arises, when the MQ scheduler selects an organization task to commit, the task is eventually canceled due to other agent's failure of commitment to other related partial process. Consequently, the agent loses an opportunity to commit to an outside task instead.

With these experiences, our next experimental setup produces the situation where there is only one agent receives two types of tasks while others are completely dedicated to the virtual organization. Even though there are five members in the virtual organization, we are only interested in the one with the multiple types of tasks. In order to reduce the noises introduced by the randomness of the occurrences of tasks, we control the frequency of tasks so that they occur regularly with a fixed time interval between two sequencing tasks. For example, a task with a frequency of 40 means there is a new arrival in every 40 clock cycles. This setup would allow us to concentrate on the effect of mapping function on the particular agent's behavior.

For this set of experiments, each run lasts 800 clock cycles. We used the following parameters: $(40, (20, 30], (0, 10], a \cdot \frac{1}{b} \cdot (1 - ((1-b)^{\frac{1}{c}})^{x}))$ for the organization task, assuming the agent has promised 20 commitments to the virtual organization and each organization task has a fixed reward of 1000. The value of a is calculated by the sum of expected reward plus the possible penalty for doing no organization task (i.e. when there is a linear penalty of 500 for each organization task, a is equal to 300000, with the expected reward 1000 * 20 = 20000 and a penalty of 500 * 20 = 10000). We varied the frequency and the reward of the outside task in order to investigate the effect of the mapping functions. In addition, we also varied the value of b in the mapping function in order to observe the behavior of the agent under both linear and progress-based penalty policies. Also in our experiment, we tried two other approaches. One is called "no mapping function", where no mapping function was used for the organization task, instead, the agent assumes a fixed monetary reward of 1000 for each organization task. Another approach is called "forced function", which means that before the agent accomplishes all its commitments to organization, it is forced to select the organization task whenever there is a conflict.

The agent has a fixed capacity of 20 tasks under our experiment settings. Figure 7 shows the number of completed organization tasks using different mapping functions, the frequency of outside tasks is 1/40, 1/20, and 1/80 in case (a), (b) and (c) respectively.

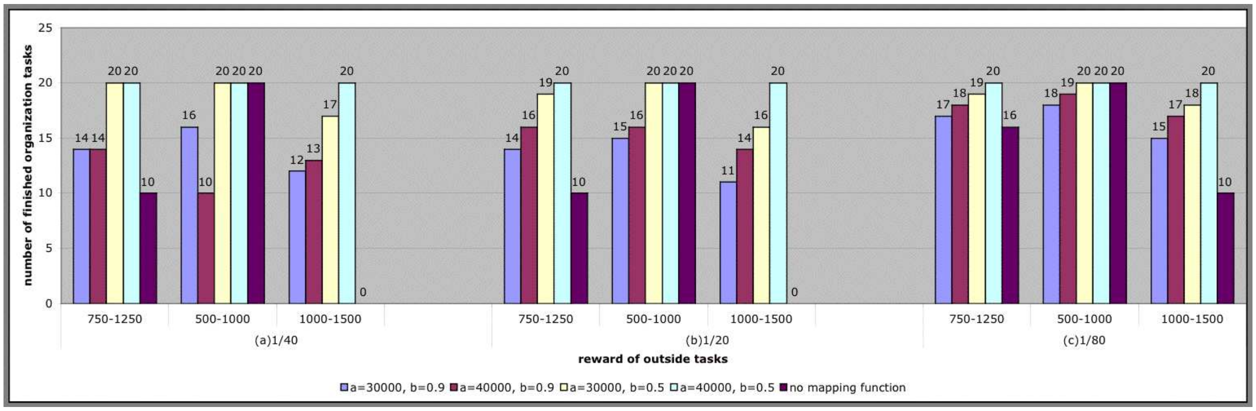

Figure 7: Effect of mapping functions on agent's promise to virtual organization

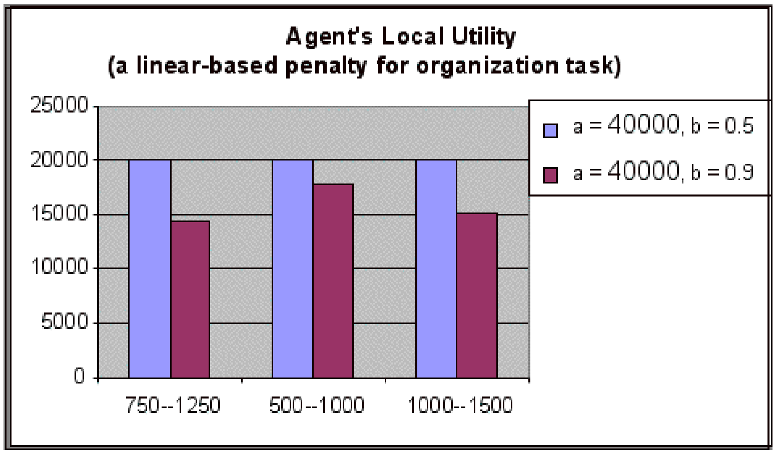

Figure 8: Agent's local utility under a linear penalty policy

In case (a), with a reward range of 500-1000 at a frequency of 1/40, an outside task is not as favorable as an organization task; therefore the agent was able to fulfill most of its promise. On the other hand, when the outside task has a reward between 1000-1500, it becomes more competitive; consequently, fewer organization tasks were being fulfilled. The most interesting case is when the outside task has a reward of 750-1250. Under this situation, the mapping function with b = 0.9 does not guaranteed the fulfillment of its promise. With b = 0.9, the agent has the tendency of favoring the first few organization tasks but then loses its momentum afterward. By contrast, when b= 0.5, it is most likely that the agent would fulfill its promises as it keeps its favor for organization tasks for longer time. Additionally, a bigger a value also reflects more emphasis on organization tasks and hence results in more commitments fulfilled. In case (b), when there are more outside tasks (the frequency is 1/20), the result is similar as in case (a); the agent performs almost the same number of tasks for the organization when using mapping functions as in case (a), this shows that the mapping function has a positive influence on the agent to keep its promise to the organization despite the increase number of outside tasks. In case (c), when there are less outside tasks (frequency is 1/80), more tasks are performed for the organization because the agent has more time available.

As described in section 3.3, we may modify the b value in the mapping function in response to the organization's penalty policy. Given a linear penalty policy, we feel that a b value close to or smaller than 0.5 would be appropriated, whereas a progress-based penalty policy may require a b value close to 0.9 to ensure that the agent would at least perform the first few organization tasks. By this convention, we would be able to control the agent's behavior in response to different penalty policies. Figure 8 shows the agent's local utility using different mapping functions, when the organization adopts a linear penalty policy (a penalty of 1000

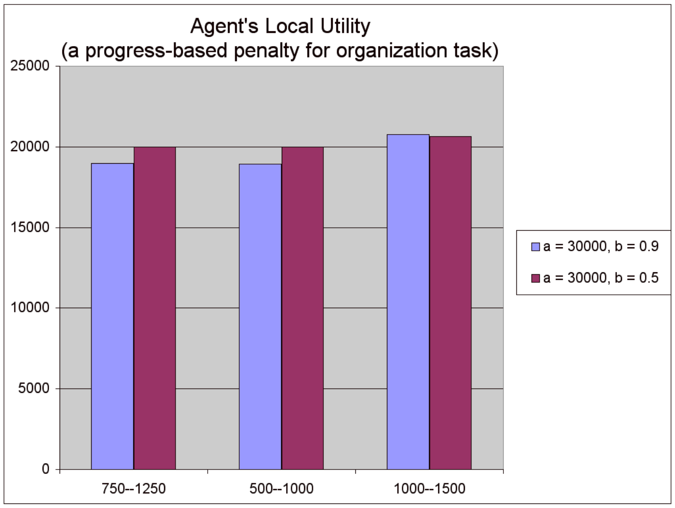

Figure 9: Agent's local utility under a progress-based penalty policy

units for each unfulfilled commitment). It is found that the mapping function with $b = 0.5$ brings the agent more local utility since it reflects the organization's penalty policy appropriately. Figure 9 shows the agent's local utility using different mapping functions, when the organization adopts a progress-based penalty policy (a penalty of 800 units for each unfulfilled commitment if total number of commitments is less than 10, otherwise, 200 units penalty for each unfulfilled commitment). Under this situation, it is found that the mapping function with $b = 0.9$ only outperforms the other one when the outside task generates higher reward (1000-1500). In the other two cases, the performances using these functions are close. This illustrates that we need a finer turn of the parameter value to reflect the progress-based penalty policy more accurately.

The use of mapping function also has a direct effect on the overall performance of the virtual organization. Since the organization utility is attributed to how cooperative the agents are, the more weights the agent put on the organization task (reflected in a and b value in the mapping function), the more organization tasks will be completed. As illustrated in Figure 10, a greater a value and a smaller b value in the mapping function have positive effect on the organization utility.

5.3 Mapping Function For Agents With Multiple Organization Memberships

Our third set of experiment was to study the effect of MQ mapping function when the agents are involved in multiple virtual organizations. We used the same settings as the previous experiment, with the parameter $(40, (20, 25], (0, 5], (750, 1000))$ for the outside task, and assuming the tasks from the two virtual organizations have the same frequency, duration, and slack time as the outside task. And once again, we focused on one particular agent who has multiple organization memberships, while others were engaged in only one virtual organization and completely dedicated to the organization tasks. Table 3 shows the agent's commitments to different types of tasks when using different mapping functions.

As indicated by the experiment result, the mapping function with a higher expected utility (a value) or a smaller b value would ensure the agent's fulfillment of its promise to Organization A. While the tasks from Organization A may appear to be more attractive than that of Organization B (when the mapping function has greater expected utility or a smaller b value), however, this does not mean the utility gain from Organization A always exceeds that of Organization B. In fact, once the agent has accumulated a certain amount of MQs from Organization A, tasks from both organization may take turns to be selected.

From Table 3, it appears as the changes in the mapping function of Organization A have no effect on the number of tasks completed for Organization B. This is because both organization tasks are also competing with the outside task. Given a capacity

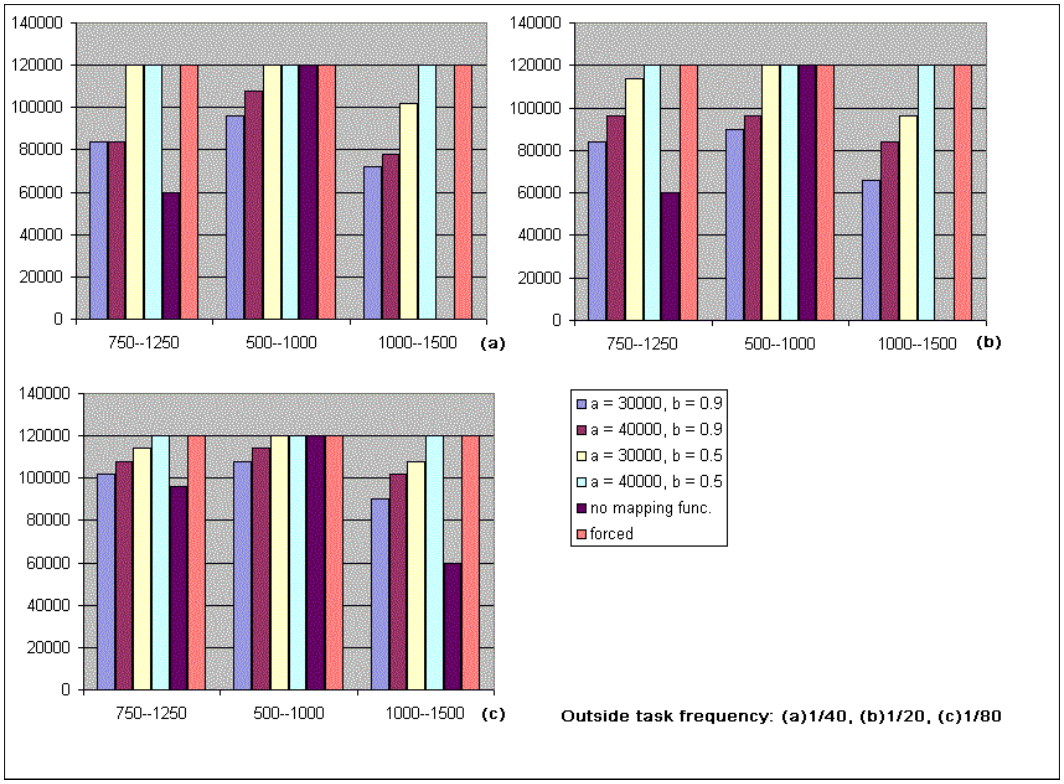

Figure 10: Organization's utility using different decision-making policies

Table 3: Effect of Mapping Function in Multiple Virtual Organization

	Organization A tasks	Organization B tasks	Outside Tasks
A&B: a=30000, b=0.9, c=20	14	14	47
A: a=30000, b=0.5, c=20 B: a=30000, b=0.9, c=20	26	13	36
A: a=60000, b=0.9, c=20 B: a=30000, b=0.9, c=20	20	14	41
A: a=60000, b=0.5, c=20 B: a=30000, b=0.9, c=20	45	13	17

of 75 tasks, there is room for the agent to achieve at least 13 tasks for Organization B. If no outside task is considered, a change in the mapping function of one task type certainly would have an effect on the completion of another type of task. When the MQs from the two virtual organizations have the similar mapping functions, there would be an equal chance for these tasks to be accepted. In this case, it may imply that an agent could not fulfill its promises to any of the virtual organizations if its capacity is limited.

One of the recurring issues we have encountered is that few organization tasks get completed when the agents are engaged in both outside task and organization task, it is even more apparent when the agents belong to multiple virtual organizations. This is the reason why the experiments are focused on one particular agent while others are assumed to have 100% availability to the organizations. When all agents are free to join any number of virtual organization, the control power of the MQ mapping function is greatly reduced because the other agents' commitments are difficult to predict. The more immediate questions are, should the initiator agent allow agents to have multiple memberships; do multiple memberships give an agent a greater utility gain when the commitments from other agents are very uncertain?

Whether an organization task can be completed is based on how dedicate the agents are. The initiator agent accepts a building task only when there is at least one agent willing to commit to each partial process/subtask, which turns out to have a probability of $\prod_{i=1}^{n} Pcommit_i$. If a virtual organization has 5 partial processes with one member agent assigned to each partial process, and suppose each agent has a probability of 0.7 to commit to the partial process, then there is only a small chance of 0.16 that a task arrive at the initiator will actually be accepted and performed by the virtual organization.

It is clear that in considering the formation of a virtual organization, one must assess how dedicate a member is to the operation of the virtual origination. The ideal virtual organization is the one that every member enterprise has 100% availability to the organization; in this case, the probability of commitment would be 1 (assuming no conflict with a prior organization task). Aside from selecting members with the higher availability, an initiator should also try to limit the number of partial processes as much as possible. An alternative solution to these limitations is to have multiple agents that are capable of one partial process in a virtual organization. However, the fact that more agents involves in an organization increases the communication/operation overhead, as well as decreases the overall profit to the initiator agent. Therefore, the initiator needs to have a balanced number of participating enterprises and partial processes in order to achieve acceptable probability of commitments from agents.

5.4 Summary of Experimental Results

From this experiment, we can make the following conclusions:

1. For an isolated instance of task selection, or in situations where there are less concern with the previously committed task, the motivational quantities framework provides an powerful tool that enables an organizationally situated agent to make intelligent decision. By considering the most important factors, an agent can reason about every aspect of its actions, thus achieve its organizational goals in a rational manner. This mechanism, however, is weaker when previously committed tasks are interfering with the current decision-making.

2. The mapping function has an effect on the agent's promise to the virtual organization, its local utility and the performance of the virtual organization. An agent could change its attitude toward the virtual organization by changing the parameters in the mapping function. For instance, an agent may react to the organization's penalty policy by adjusting the parameter values in the mapping function, thus change its attitude as to how to fulfill its commitment to the virtual organization. It is possible to formulate a mechanism to calculate the value of the parameters in the mapping function in order to reflect the organization's penalty policy accurately.

3. An important issue faced by the initiator agent is how to achieve a balance between the number of participating enterprises and the number of partial processes in order to achieve acceptable probability of commitments from agents.

6 Related Work

The research on organization and intelligent agents has been conducted by many other researchers too. Bond [2] has proposed a computational model for organizations of cooperative agents, which captures properties of relationship and organization in sets of distributed intelligent agents. It introduced a concept of commitment, which represents mutually agreed constraints on action, belief and world state. [3] proposed a conceptual framework for agent societies, consisting of three interrelated models, that distinguishes between organizational and operational aspects of the domain. Contract rules specify commitments between agents and society concerning role enactment, and commitments between agents concerning interaction. We use a similar commitment concept in our work but we are more focused on how to motivate the agents to keep their promises.

Dynamic organization of multi-agent systems has been studied. [11] discussed self-adaptation of organizations in multi-agent systems according to the dynamic of interactions between agents. Starting from a default organization, the architecture of acquaintances evolves autonomously depending on messages flow in order to improve the global behavior of the system. It proposes three principles that can be applied to adapt the organization: ”have a good address book”, ”share knowledge”, ”recruit new able collaborators”. [14] studied self-organization of agent systems through bottom-up coalition formation, [7] described how to use self-diagnosis to adapt organizational structures. Virtual organization can be viewed as a dynamic organization structure, however it adopts a totally different approach.

[1] presented experimental results that show no one decision-making framework performs best across various situations that may be faced at run-time. Agents who implement the capability of Adaptive Decision- Making Frameworks (ADMF) are able to dynamically modify their decision- making frameworks at run-time to best meet the needs of their current situation. This is also supported by our experience in this work - agent needs to dynamically adjust their local decision-making procedure in order to best fit with the organization’s context and the environment.

Multi-Agent systems have been used to simulate different types of organizations. [6] used a multi-agent system to model a set of firms in competition with each other within a shared market. [5] presented an approach towards process-oriented collaborative inventory management in supply chains, taking advantage of multi-agent technology in terms of modeling and simulation. [10] has studied market-based approaches for task-assignment multi-agent systems, it empirically evaluated these organizational forms according to the amount of communication required and the rate of failed task-assignments, and compared them to a system without organizational forms. Our work has a different emphasis from the above work, it is focused on the decision-making process and the influence of each individual’s decision on the organization.

7 Conclusion and Future Work

With the changing landscape of business world, cooperation between the enterprises is the only way to stay on the edge of competition. Cooperation enables enterprises to share skills, costs, access to one another’s markets and resources and, at the same time, decrease the risk of investments. Supported with the rapid development of information technologies, virtual organization has the potentials to be the future way of enterprise cooperation and electronic business.

This paper investigated the challenges and obstacles that we are still facing. We proposed a negotiation protocol for automatic formation of a virtual organization. We have studied the decision making of individual agent in a multi-dimensional negotiation process. The partner selection process is another issue that we have focused on, we presented a RBFS algorithm to find the optimal membership for the virtual organization. This solution we applied to Virtual Building Company may not be adequate for a large number of agents and bids, some sort of heuristics or filters are needed for the screening of bids in order to reduce the complexity. We have incorporated the motivational quantities framework for the task selection process so that agents can make rational decision during their operation. We presented a utility mapping function that can model the agent's preference, promise and penalty policy of the organization. We adapted a statistical model that allows us to predict and analyze the agent's behavior and the influence on the organization utility. We have also attempted to study of the agent's local control - how the utility mapping function of the MQ affects the local time/resource allocation and the agent's overall utility achievement.

Though we have gained a big picture view of virtual organization through this study, we also find there are areas where further studies are needed. For example, we have not explored how the agent make decision in the initial bidding process, how the agent should decide whether to bid and how to bid. Analytical work also needs to be done in the statistical model to study how to adjust the parameter in the mapping function in order to optimize some organizational objectives. These, of course, would lead us to our future work.

8 Acknowledgment

The authors would like to express their thanks for Victor Lesser for the permission of using MASS and JAF software. Also thanks to Tom Wagner for his permission of using the MQ Scheduler. Thanks Jiaying Shen for discussion related to the statistical model.

References

[1] K. S. Barber and C. E. Martin. Dynamic reorganization of decision-making groups. In *Proceedings of the fifth international conference on Autonomous agents*, pages 513–520. ACM Press, 2001.

[2] A. H. Bond. A computational model for organizations of cooperating intelligent agents. In *Proceedings of the conference on Office information systems*, pages 21–30. ACM Press, 1990.

[3] Virginia Dignum, John-Jules Meyer, and Hans Weigand. Towards an organizational model for agent societies using contracts. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 694–695. ACM Press, 2002.

[4] Ana Paula Rocha Eugenio Oliveira. Agents advanced features for negotiation in electronic commerce and virtual organisations formation processes. In *AgentLink 2001*, pages 78–97. AgentLink, 2001.

[5] Yonghui Fu, Rajesh Piplani, Robert de Souza, and Jingru Wu. Agent-based simulation applications: multi-agent enabled modeling and simulation towards collaborative inventory management in supply chains. In *Proceedings of the 32nd conference on Winter simulation*, pages 1763–1771. Society for Computer Simulation International, 2000.

[6] Z. Guessoum, L. Rejeb, and R. Durand. Multi-agent simulation of firms and organizational forms. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 1000–1001. ACM Press, 2003.

[7] Bryan Horling, Brett Benyo, and Victor Lesser. Using self-diagnosis to adapt organizational structures. In *Proceedings of the fifth international conference on Autonomous agents*, pages 529–536. ACM Press, 2001.

[8] Bryan Horling and Victor Lesser. A Reusable Component Architecture for Agent Construction. Computer Science Technical Report TR-98-30, University of Massachusetts at Amherst, May 1998.

[9] Bryan Horling, Regis Vincent, and Victor Lesser. Multi-agent system simulation framework. In *16th IMACS World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation*. EPFL, August 2000.

[10] Tore Knabe, Michael Schillo, and Klaus Fischer. Inter-organizational networks as patterns for self-organizing multiagent systems. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 1036–1037. ACM Press, 2003.

[11] P. Mathieu, J. C. Routier, and Y. Secq. Dynamic organization of multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 451–452. ACM Press, 2002.

[12] Daniel E. O'Leary. Virtual organizations: two choice problems. In *Proceedings of the international conference on Information systems*, pages 145–154. Association for Information Systems, 1998.

[13] Jiaying Shen, Xiaoqin Zhang, and Victor Lesser. Degree of Local Cooperation and its Implication on Global Utility. *Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, July 2004.

[14] Mark Sims, Claudia V. Goldman, and Victor Lesser. Self-organization through bottom-up coalition formation. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 867–874. ACM Press, 2003.

[15] Thomas Wagner and Victor Lesser. Relating quantified motivations for organizationally situated agents. In N.R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI (Proceedings of ATAL-99)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000.

[16] 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.