A STUDY OF A DYNAMICAL NEGOTIATION ARCHITECTURE FOR MULTI-AGENT SYSTEMS

Chuan-Tsai Lin and Alan Liu

Department of Electrical Engineering,
National Chung Cheng University
Ming-Hsiung, Chia-Yi, 621, Taiwan
Email: aliu@ee.ccu.edu.tw

ABSTRACT

We discuss about problems which occur during the negotiation process of multi-agent systems (MASs). In the MASs, agents may have different skills and expertise to solve problems. The negotiation process begins when conflicts occur during the cooperation of agents. We propose an approach for choosing a solution among different solutions provided by different agents. Our approach uses a hierarchy to represent the reason and its weight associated to it. This paper presents solving the problems of negotiation in MASs by using dynamic negotiation architecture. There are two weights proposed in this paper. They are the task weight and the property weight. These two weights help us make decision about taking a solution from the conflict solutions. A task hierarchy and a property hierarchy are used for the calculation of these two weights.

1. INTRODUCTION

A multi-agent system (MAS) may have agents which are across multiple disciplines. There may be conflicts in the solutions provided by these agents from different domains. With these conflicts, the solutions from these agents become unable to apply to solve problems, and usually we do not know how to make decisions between these solutions. If they can cooperatively solve problems and resolve conflicts by negotiation, it would be of great help for solving complex problems. In this paper, we discuss the negotiation issues in the cooperation. To resolve conflicts, we propose a negotiation approach in this paper.

In this paper, we propose to use a hierarchy containing different information to help solve the negotiation problems of the multi-agent system. The main idea is that, two weights are used for the decision making process. These two weights are the task weight and the property weight, which are calculated from the task hierarchy and the property hierarchy.

In designing our method, we keep in mind what Lesser proposed as the three key principles that can be used in building multi-agent systems [1] --- the criteria for performance, the flexibility of resources, and the efficient cooperation method.

In Section 2, we survey the cooperation and negotiation approaches from other papers. In Section 3, we propose our approach for negotiation and give a comparison between our approach and others. In Section 4, a manufacturing system, which uses a multi-agent system as its underlying architecture, is presented as the examples of our approach. In Section 5, we give a conclusion.

2. NEGOTIATION ISSUES IN AGENTS

The main principle of conflict resolution is by negotiation. The meaning of negotiation is to change the primitive plan, task allocation and resource allocation by communication. It is a method of solving conflict in the situation of cooperation [13]. An agent only has local view, and there may be a limit in resources and skills. It causes a conflict when agents have different actions on common things. For example, the expert of forestry recommends a forest on travel for your health. However, another expert of education suggests you to a museum for get knowledge. There is not a fault in the two experts, but who should we agree it? Thus, it should be a negotiation to make decision. Negotiation is needed in the follow situations [2,3].

1. Agents have different opinions among them.
2. Agents want the same resource simultaneous.
3. Agents have the different goals of some phases.
4. Agents does not know why others intention.
5. The cost of solving conflict is higher than solving problem.
6. An agent opposes the behavior of another and demands a concession of competitor.

There are many methods of negotiation published by others. For instance, Robinson [4] believed that negotiation is a part of specification. A negotiation knowledge should be used it for bargaining. It gives the value of satisfaction from 0 to 100% with reference to each suggestion of agent. Tan, et al. [5] also proposed to regulate a range of value for requirements and preferences of customers and finish the conflict detection and conflict resolution by referring to the range of value and then use it in concurrent engineering. Kakehi, et al.[6] proposed to solve conflict by a negotiation protocol. They establish a protocol of negotiation to solve the conflict of common resources. When an agent finds a conflict with another, they will get into negotiation process. If the resource of their conflict is a reusable resource, the agent informs another about his intentions by sending messages and to finish their actions in a compatible method. However, If the resource of their conflict is a consumable resource, they try to search a replacement of the resource. When they find the replacement, the original finder concedes the original resource and to get it. If they cannot find a replacement by themselves, they search replacement for competitor. If they find it, the competitor has to yield the original resource and they all finish their tasks. If they cannot find any replacement, it will make decision by tossing a coin. Krovi [3] proposed a virtual negotiation environment each kind of restriction and show their strategy is effectively by simulate method. Hu, et al. [7] proposed to establish a blackboard. When an agent has questions, it posts message in blackboard to inquiry other agents. Then other agents post their suggestions. After the sender sort suggestions, it selects the best and reply to blackboard. Other agents also reply approval or the reason of disapproval.

Among the methods of negotiation that are proposed by researchers, we focus on the following structures [8,9]:

1. an arbitrator making decisions after negotiation between agents,
2. all agents participating in negotiate directly with each other,
3. an environment like blackboard to help agents in negotiation.

It is a problem that who the leader is in the cooperation in multi-agent. We have surveyed two methods. The first is a constant leader, who is responsible for receiving tasks and distribute it to other agents. The advantage is its simplicity, but the disadvantage is that the leader has much more workload. If the leader breakdown, all systems will shut down together. The second is that each of agents has the ability to become a leader. They can receive great task and find other agents to help finish it. Each agent knows the specialty of other agents. The advantage of this type is that it conquers the shortcoming of first type. It distributes the workload of single leader. When an agent breaks down, there are other agents to replace it. The complexity is the disadvantage of the second type. Because every agent can be a leader, it should have much information about others in each agent and they must have ability of assigning work to others.

In our research, we have two kinds of agents that are active agents and passive agents. The active agent has data about specialty of others, and there is a task hierarchy in the active agent. The concept of task hierarchy contains the specialty of each agent in the environment and the different weight of agent when executing different tasks. When a new agent is added in this environment, all active agents can reload the information of the task hierarchy simultaneously. The passive agent cannot be a leader. Therefore, it cannot find others to give a service actively. It only provides help to user or other agents. This agent is the same as a general passive agent mentioned in [10]. The detailed discussion will be presented in the next section.

3. SOLVING PROBLEMS OF NEGOTIATION

In our daily life, we might resolve conflicts using the following approaches: vote, determined by a chairman, drawing, taking turns. In the same idea, we integrate these approaches to propose our approach. We use a task weight to denote the social position of agents. The social position means the priority of roles in the environment. When there are many agents in an environment, we use this information to know which has higher status and which has not. The task weight of an agent is dynamic and changes with different tasks. When we construct each agent in the multi-agent system, we construct a hierarchy according to their abilities and their skills. The information, rules, etc., everything in its mental state, which involves with the behavior of the agent, have a hierarchy. In different level of the hierarchy, the weight, which we call it as property weight, is different. The property weight dynamically changes with the task, which the agent is taking, and the properties, which are emphasis by the user in the task. We use task weight and property weight to resolve the conflicts.

If we want to solve the problems of negotiation, it must have the negotiation knowledge. An important part of negotiation knowledge come from requirements analysis. In this paper, we propose another important part of negotiation knowledge that comes from more information which is represented as different hierarchies.

3.1 Task Hierarchy

Each agent is assigned a task weight, which is the result of the analysis of requirements. Each agent plays a role in an MAS. When the problem is cross multiple domains, the cooperation of experts from different domains is needed. The task weight is to determine the importance of an expert of a specific domain to the task, which the MAS wants to accomplish. With a higher task weight, the expert has a higher social position and takes more important role compare to an agent with a lower task weight. The task weight of an agent changes with different tasks.

We construct the agent task hierarchy according to the agents, which can be achieved by the agents. There are three levels in this hierarchy. The top level is task level, which is a classification and an abstraction of the tasks, which can be achieved by the MAS. The next level is the sub-task level, which fulfills the detail parts of the abstract tasks. The lowest level is the agent level, which denotes the abilities of an agent. As illustrated in the Fig.1.

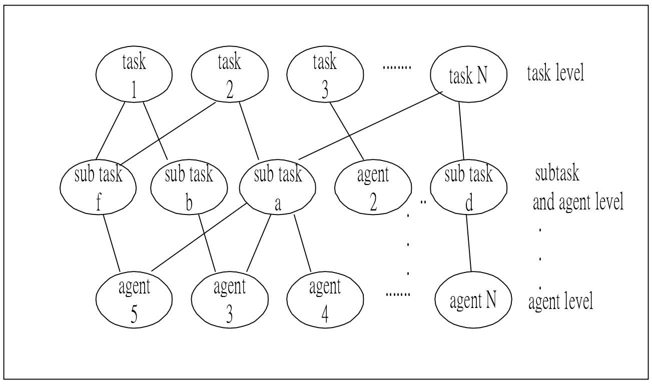

Figure 1. Task hierarchy

In Fig.1, Task 1 is composed of sub-Task f and Sub-Task b. Agent 5 has the ability to accomplish sub-Task f and to accomplish sub-Task a. Agent 2 has the ability to accomplish Task 3. The middle level of this hierarchy can be partition into more than one layer according to the functions of the MAS. The agent at lower position on the hierarchy can accomplish the task or sub-task above it. From a task's point of view, an agent, which can accomplish the most amount of the task's sub-task or can accomplish the entire task, has the highest task weight. To different task, every agent may have different agent task weight.

We construct the task hierarchy by the assistance of human expert. Human experts analyze the requirements of task and sort them by their importance. Afterward, they arrange the tasks and subtask a suitable node position. According to the ability of the agents, give them a node position. We can devise some works and access the ability of agent by the result of how it does those works. We compare the result with our standard to judge the agent's ability. The standard comes from user requirements and the result of human experts doing the work. According to the ability of agent, we adjust its position of the hierarchy.

We construct the system according to the kind of tasks that we can execute to establish the task level. Afterward, according to those subtasks of the task or the agent has ability about the task to construct the subtask in the agent level. In the light of the relations is close to or not to establish the task hierarchy in order. The last level of hierarchy is the agent level in which an agent has functions about the task of his upper layer. When we add new agent in this system, he has ability to place it under those task according to the new task or original task. If the distance between them is shorter, that means the relation of them is close. Therefore, after we finish the construction of the system, the task hierarchy is fixed. Unless adding new agents or tasks, the task hierarchy will not change. When we compute the task weight of each agent, it is based on two essential factors. One is about the task or subtask above the agent that we give it the rate of weight. If we respect for the task or subtask, we give them higher weight. The other factor is the distance between agent and the task. If the distance is shorter, the weight is higher. The two factors decide the weight of every agent.

Fig.2 shows an example of the task hierarchy. It contains a task hierarchy of a traveling system and a manufacturing system. The traveling system concerns about the trip's purpose, safety, and cost; whereas the manufacturing system focuses on the cost and specification of a product.

3.2 Property Hierarchy

When we construct each agent, we construct a hierarchy according to the abilities of each agent and the property hierarchy. By "property", we mean that is an entity of user requirement, such as safety, cost, etc. Under the first level, it is the category of behavior or information that may influence the property. It is the category level. Every category will be constructed under the property that has some relation to each other. If the relation is weaker, the distance is longer. The last level is the behavior or information of the agent. It has some relation with the property or category of its upper layer. The architecture is a property hierarchy of agent.

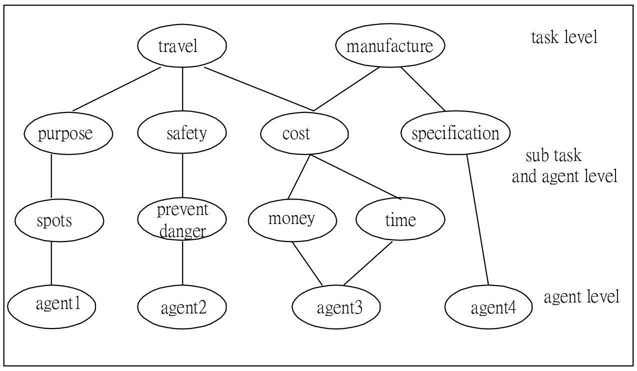

Figure 2. Example of the task hierarchy

Therefore, after constructing an agent, it completes its property hierarchy at the same time. When executing an agent, it will give all property different value of weight by user's respect. Afterward, we want to give each behavior and information different weight, according to the shortest distance between it and the property. We call this weight the property weight. The different task may produce different property weight. For the reason, the property weight is able to change dynamically. It will influence the result of agent negotiation.

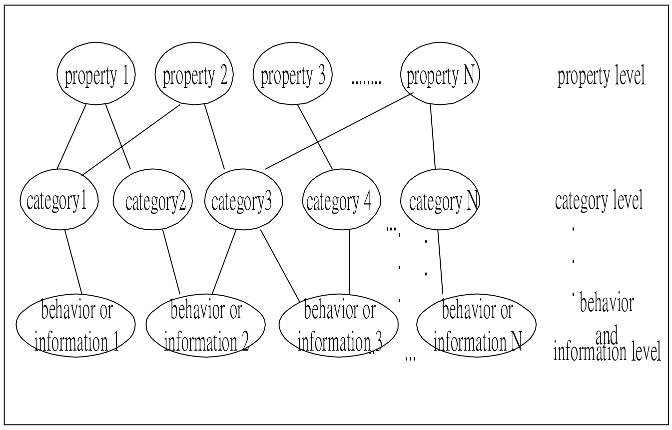

Figure 3. Property hierarchy of an agent

The node, which is of a high level, has higher position. The hierarchy can also be a flat linking model. In the Fig.3, the property level denotes for the property of the goal which we want to accomplish. Like safety, free, efficiency and simple, these are kinds of properties belong to certain goals. The lowest level, the behavior and information level, of this hierarchy denotes the behavior of an agent or the information provided by the knowledge base. The category level is the level of the state transition of the behavior and information level to the property level. This level can represent as a single layer or multiple layers. In this hierarchy, the behavior of an agent and the information provided by the knowledge base is derived from the properties related to the behavior or information in the hierarchy. The more few nodes between the property and the behavior or information, the more strongly property and the behavior or information are related. When our main interest is a specific property, which is on the property level of this hierarchy, the behavior and information related to this property is of importance, which we denote as the property weight.

When we want to construct an agent, we should have to construct the property hierarchy of the agent. We construct the property hierarchy by the assistance of human expert. Human experts analyze the abilities of the agent and classify them. Afterward, they arrange the categories, behaviors and information in a suitable node position. After using this agent some times, we can change suitably the node position by the results of test. The method of how does we adjust its position that is alike as the method of the task hierarchy.

We also allot different degree of importance to every level. To distribute two states of finish tasks as follows. One is the *complete achievement* and the other is the *acceptable achievement*. The complete achievement means the consult is completely conformed to the requirements of each agent. Therefore, the output of consult does not need the process of negotiation. At the same time, the acceptable achievement means that the result does not completely fit in with the requirements of agents, but the part of accomplishment is the most principal and the result is acceptable. It means the method is a compromise. When its requirement does not be accepted, it has some pointers to help to influence the property weight. After changing property weight and negotiation, the result may also be fit with our requirements. In spite of the result of not completing to conform to our requirement, it is acceptable. Thus, it can avoid endless negotiation to find a complete solution that is not existence.

To set the value of the property weight and the task weight, we require the degree of interest to task and sub-tasks, and the degree of interest to properties. The degree of interest to task and sub-tasks is used to determine the value of the task weight and the degree of interest to properties is used to determine the value of the property weight. These values become the basic value of the task hierarchy and the property hierarchy. In these hierarchies, the nodes, which are far away from the task and properties which are most interested by the user have lower weight than nodes which are close to the task and properties.

After users input the values in their requirements, the values are processed by the method of pair-wise comparison using eigenvalue and eigenvector [11,12]. These results become the weights of the nodes in the hierarchy. When we input any ratio of nodes that are different entities of user requirements, this method can obtain some normalized values that are the weights of nodes. The sum of those normalized values is equal to 1. We base on these normalized values to extend the weights to other nodes. Fig.4 shows the model of input analyzing using pare-wise comparison. The inputs are some values of nodes and the outputs are the weights of these nodes.

3.3 Negotiation Method

We use the task weight and property weight to make decisions to choose a better solution. The task hierarchy and the property hierarchy determine these two weights. We give different proportion of power of determination to agents with different task weights to finish a task.

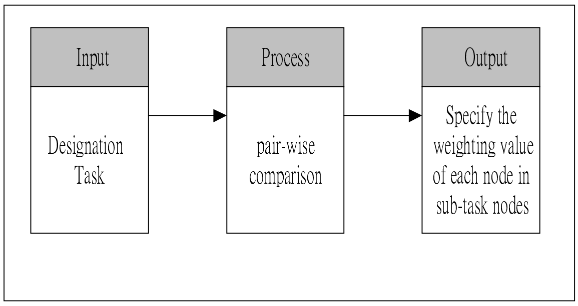

Figure 4. Input analysis using pair-wise comparison

Because those agents who deal with the task may have conflict by different views, if it will select the suggestion by their position, we will take the suggestion of the agent of a high task weight. However, sometimes the agent of a lower task weight proposes a suggestion that is important for its view, and the agent of a higher task weight proposes a suggestion that is not respected for its view. Therefore, we should consider the property weight and the task weight to make decision together to influence the result. The property weight presents the respect degree of suggestion of each agent proposed. Furthermore, if an agent insists on its suggestion, it may increase the property weight by physical situation. The physical situation means it is according to information about “designation task”. There is some negotiation knowledge to help to change the property weight.

When an agent, say the agent A, receives a task, which the agent A cannot accomplish alone, the agent A will find other agents to cooperate to accomplish this task and the agent A will become the leading agent of this task. All agents receive a part of this task from the leading agent and accomplish the sub-tasks. After they accomplish their subtasks, each agent checks the result of global integration of their solutions. If there are conflicts occurring, the negotiation process begins.

When an agent is not satisfied with the result of the global solution, which has conflicts with the solution proposed by this agent, this agent gives the property, by which its proposal is made, to the leading agent and asks the leading agent to reconsider about the solution. The leading agent bases on the concept of “designation task” to change the property weight of agents. Afterward, find the solution of the maximum weight and output after all agents adopt it. The meaning of all agents adopting it is when the behaviors of related agent check the solution and cannot find conflict.

For an example, if there are two agents, say Agent 1 and Agent 2, have conflicts in their solution. The proportion of their task weight is Agent 1:Agent 2 = 5:3. Agent 1 gives the proportion of property weights Solution 1: Solution 2 = 2:3. Agent 2 gives the proportion of property weight as Solution 1: Solution 2 = 4:5. We take the solution 1 because 5*2 + 5*4= 30 is greater then 3*3+3*5 =24. If Agent 2 refuses to accept the solution proposed by Agent 1, Agent 2 has to provide the reason. In another word, Agent 2 should provide the behavior of conflict, then the leading agent changes the property weight by the effect of the behavior and its property.

We can see from above, the agent with a small task weight still has opportunity to determine the solution if this agent has a great property weight. In other words, the solution proposed by an agent with a great social does weight and a small property weight, may be overthrow if there is an agent, which does not agree with this solution and provides a reason. The reason is about that the solution does violate some of properties, and those entities of user requirements are important.

3.4 Negotiation Architecture

We proposed the architecture of negotiation, which is different to these architectures we survey in the previous section. In our architecture, when there are conflicts, there is a leading agent to lead the process of negotiation. The leading agent negotiates with each agent, which has conflicts, to understand the reasons of the conflicts and then the leading agent makes a global assessment and adjusts the global solution. Again, the leading agent asks other agents if they are satisfied with the global solution or not. The leading agent iterates the negotiation process. We use the Fig.5 as an illustration. When the conflicts occur, the negotiation process begins. The leading agent will find out agents which have conflict solutions and then negotiates with these agents individually to understand their reasons to make conflict solutions. The leading agent collects the reasons from these agents, which have conflict solutions, and compares their weights and the reasons they provide to make decision of which solution to take. If all agents agree with the global solution, the negotiation process ends; otherwise, the leading agent iterates the negotiation process.

This approach is similar to the approach of using a moderator. Our approach is different from the approach of using a moderator in the way that the leading agent in our approach considers about the ideas and reasons provide by other agents and integrates these ideas and reasons to make its decision but not to arbitrate. With different tasks, the leading agent of our MAS is changeable, which makes our approach more flexible.

To use this system, we have to use one of the agents, which have the ability to become a leading agent, and input our requirement (data). The leading agent analyzes the input data and find out other agents to cooperate to carry out the task. The task weights and the property weights are gained after the data analysis. When conflicts occur, the leading agent compares the weights and dynamically adjusts the property weights to resolve conflicts and thus carry out the given task. The sequence of user involvement can be described as follows. First, the user enters the task request. Afterward, the active agent becomes the leading agent of this task. It will find other agents to help and the system will propose some questions about detail requirement of task. After the user answers the questions, the system will arrange negotiation knowledge according to the answer. The negotiation knowledge contains the task weight, the property weight and the regulative standard about those weights. We can adjust the property weight of agents by referring to the regulative standard. The regulative standard is the value that comes from designation task. Those agents propose the suggestion about solutions and output the most suitable solution.

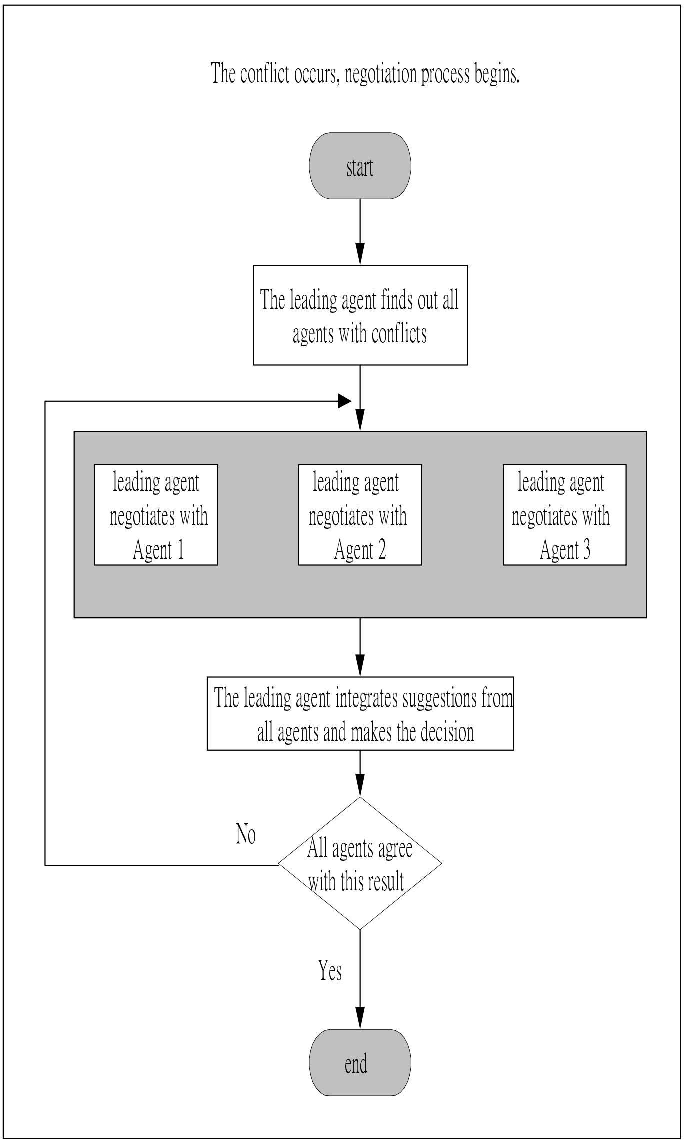

Figure 5. Negotiation process

3.5 Comparison with Other Methods

Wong [9] proposed a method that gives a social position to each agent. Therefore, there are different relations between those agents. For example, the agents communicate each other with equal status, the higher status of agent communicate with lower status of agent and the vice versa. When the different roles communicate with each other, it will use different methods. Our method allocates a task weight to each agent dynamically and the agents will make decision of negotiation by their position of the hierarchy. Besides, we can solve the three conflicts as Wong's proposal and further solve the resource conflict.

Hu, et al. [7] proposed to establish a blackboard. When an agent has questions, it posts message in blackboard to inquire other agents. Then other agents post their suggestions. After the sender sort suggestions, it selects the best and reply to blackboard. Other agents also reply approval or the reason of disapproval. The process is executed repeatedly until all agents are approved. The difficulty is how to sort the suggestions. It is not correct that the best suggestion must come from best agent, and we also cannot know whom is the best agent. In accordance with their shortcomings, our method of social and property hierarchy is able to indicate a standard to solve the conflict. The standard comes from human experts and user requirements.

Table 1 shows a comparison among some cooperative methods of MASs. The arbitrator method means there is an arbitrator in the system. When we want to add a new agent in this system, we should inform the arbitrator about the ability of the new agent. The arbitrator will allot task to other agents according to the ability of those agents. The shortcoming of this method is that the load of arbitrator is higher than other agents and the system will shut down when the arbitrator breaks down. The central blackboard means that there is a central blackboard in the system. When an agent needs help, it can post message onto the central blackboard. The agents read the announcement and reply it. When we want to add a new agent in the system, it just needs to keep watching the central blackboard. An advantage is that inserting new agents requires little modification to the system. It is less efficient but flexible. Another shortcoming of this method is that it has a bottleneck of communication [7]. When the central blackboard is shut down, the system will be like the arbitrator system. The common disadvantage of the two methods is less stabilization. When our system wants to add a new agent, we should analyze the ability of this agent and add it into task hierarchy. This step is heavier than other two methods, but there is more stabilization than them.

4. EXAMPLE

Ten, et al. [5] proposed a method of design and manufacturing in concurrent engineering. Now we use our method to solve their problem. In this example, they wanted to design and manufacture a bracket that holds the camera. Three points considered are

1. Time: It should finish the wok within permissive time.
2. Budget: The cost must within budget.
3. Specification: It should fit in with our request about the length, the strength and the rotational ability.

In their approach, they had a process of design first. After design, the process of simulate manufacturing took place. If the process of simulation manufacturing failed, it should be redesigned. The scenario started from the completion of the initial design. Afterward, considering whether it corresponded with the request in all respects. The Accounting CD (Conflict Detection) agents found it exceeds budget. The local CR (Conflict Resolution) Agent was not able to solve the conflict, thus it handed over to Global CR Agent to solve this conflict. When Manufacturing CR Agent designed the task again, it solved the original conflict, but a new conflict was found by Material CD Agent. If the strength was not insufficient, Material CR Agent solved the problem and finished the process of design. When executing the process of simulation manufacturing, the Simulation Agent found that it could not be manufactured within permissive time and it should be redesigned. They eliminate the unsuccessful causes and made a new design that can finish in the permissive time. Although the design had some difference with request in specification, the variation was within the scope and it accepted.

To show our method, we repeat the example. When we want to accomplish their task by our method, we should have three agents as follow.

1. Time Agent: It manages how to the wok within permissive time.
2. Accounting Agent: It manages the cost within budget.
3. Specification Agent: It manages the specification to fit in with our requirement.

During the process of design, Specification Agent combines with some parts to design a product that correspond

Table 1. Comparison between systems on adding a new agent.

methods / item	Arbitrator	Central blackboard	Our method
A new agent is added into the system.	Inform the arbitrator	Post onto blackboard.	Add in the task hierarchy and copy to other active agents.
Restriction	1.Less stabilization than our method. 2. If the arbitrator does not have global view, it cannot get an objective solution.	1.Less efficiency than arbitrator. 2.Less stabilization than our method.	Less flexible than the central blackboard.
Advantage	1.It is simpler to add a new agent than other two methods. 2.It is easier than other two methods to solve conflict.	Flexible	1.Stabilization. 2.It is helpful to negotiation by the different roles among agent.

to the requirements. Afterward, Accounting Agent and Time Agent check whether it has conflicts in their requirements or not. If the answer is no, output the result and stop; else it enters the process of negotiation. In the negotiation process, the leading agent checks the task hierarchy to know what request the customer cares more. Is it time, cost or specification? The request that the customer cares more, the more weight the agent has. When we consider the property hierarchy, the behavior of an agent that customer cares more gains more weight. Besides, we can adjust to the weight according to the difference between requirements and finished result. After comparing the weight, the negotiation is produced.

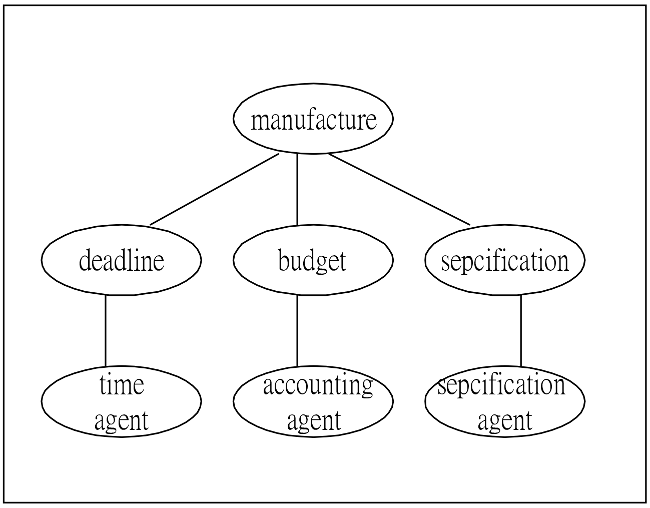

Figure 6. Task hierarchy of manufacturing system

Fig.6 shows the task hierarchy of Manufacturing System. We construct it by human expert knowledge. As the task hierarchy, if we want to accomplish the task by our method, we should have three agents. They are Specification Agent, Accounting Agent and Time Agent.

According to the task hierarchy, we should devise some questions to understand the priority of user requirements. Fig.7 shows the property hierarchy of Specification Agent. According to the property hierarchy, we should also devise some questions to understand the priority of the categories.

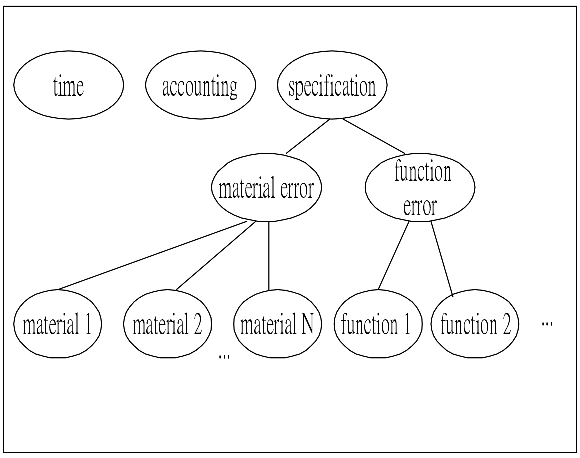

Figure 7. Property hierarchy of specification agent

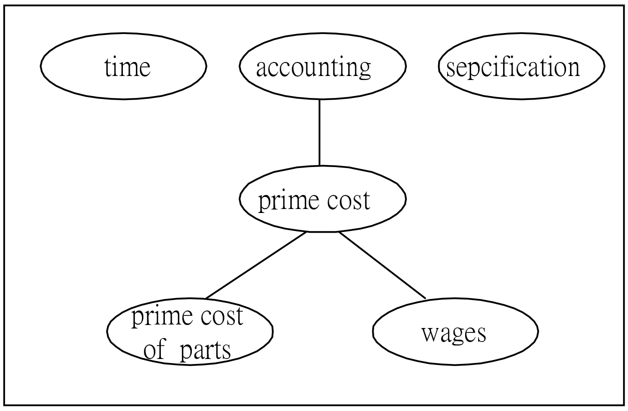

Figure 8. Property hierarchy of accounting agent

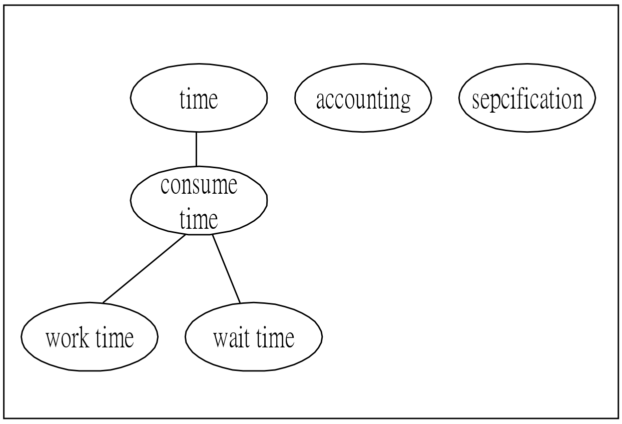

Figure 9. Property hierarchy of time agent

Figs. 8 & 9 show the property hierarchy of Accounting Agent and Time Agent respectively. Table 2 shows the specialties of each agent in Manufacturing System. Table 3 shows the content of parts in the database of Specification Agent. After finishing this system, we compare it with the application of [5]. We detect our solution is the same as theirs. Their system must have two steps to finish the task, the simulated design and simulated manufacture. However, our method only needs one step. We simulate the design process and manufacturing process at the same time. This is an advantage of our method.

Table 2 The specialties of each agent.

Name	Specialty
Specification Agent	Manufacturing product by assembling the parts. Checking material. Checking length. Checking degrees of rotation.
Accounting Agent	Checking cost.
Time Agent	Checking time.

Table 3. The content of the parts

No.	Type	Material	Length (cm)	Rota-tion	Price (dollars)	Time (hours)
1	A	Metal	20	0	30	5
2	A	Wood	15	0	25	2
3	A	Metal	12	0	40	4
4	B	Metal	15	50	50	8
5	B	Wood	25	35	25	6
6	B	Wood	30	40	5	1

5. CONCLUSION

There are four proposed concept in this paper. The first is the task hierarchy and property hierarchy. It is helpful to conflict resolution. The second is the active agent and passive agent. It is helpful to the reliability of MASs. The third is the designation task. It is helpful to the validation of the solution. After finishing these systems, we make it to execute some specific tasks. We find that the result is close to our expectation. The last is a negotiation method. It is helpful in reducing the cost of a system.

6. ACKNOWLEDGMENT

This research is supported in part by National Science Council under grant NSC 89-2213-E-194-021.

7. REFERENCES

[1] Victor R. Lesser, ”Cooperative Multiagent Systems: A Personal View of the State of the Art”, IEEE Transactions on Knowledge and Data Engineering, VOL. 11, NO. 1, pp. 133-140, January/February 1997.

[2] Katia P.Sycara, “Multiagent System”, AI Magazine, pp. 79 - 92, Summer 1998

[3] Ravindra Krovi, Arthur C. Graesser, William E. Pracht, “Agent Behaviors in Virtual Negotiation Environments”,IEEE Transactions on system, Man, and Cybernetics-Part C: Applications and reviews, pp. 15-25, February 1999.

[4] Robinson, W.N. “Negotiation Behavior During Requirement Specification” Proc. of 12th International Conference on Software Engineering, 1990, pp. 268-276

[5] Gek Woo Tan, Caroline C. Hayes, and Michael Shaw, “An Intelligent-Agent Framework for Concurrent Product Design and Planning”, IEEE Transactions on Engineering Management. VOL. 43, NO. 3., August 1996, pp. 297 – 306

[6] Kakehi R and Tokoro M. “A Negotiation Protocol for Conflict Resolution in Multi-Agent Environments”, Proc. of International conference on Intelligent and Cooperative Information Systems, 1993., pp. 185 -196

[7] Hu Yongtong, Liu Ping, Yan Yuhong, Zheng Danian, Ma Changchao, Bode, j., Ren Shouju. “ A Multiagent System For the Support of Concurrent Engineering”, Proc. of IEEE International Conference on Systems, Man and Cybernetics, 1996., Volumn:2, 1996, pp. 959-964 vol.2

[8] Werkman, K.J.; Barone, M.; Hillman, D.J.; Wilson, J.L., ” Designer Fabricator Interpreter System Evaluating Alternate Connection Configurations Through Multiagent Negotiation “, Proc. of The 6th Conference on Artificial Intelligence for Applications, pp. 153 –159, 1990

[9] Wong, S.T.C. “Coping with Conflict in Cooperative Knowledge-Based Systems”, IEEE Transactions on Systems, Man and Cybernetics, Part A, Volume: 271, Jan. 1997, pp.57-72

[10] Ferber Jacques, “Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence”, Addison-Wesly:Harlow England, 1999.

[11] Les Frair, “Student Peer Evaluations Using the Analytic Hierarchy Process Method”, Proc. of Frontiers in Education Conference, 1995, pp. 4c3.1-4c3.5 vol.2

[12] Saaty, Thomas, L., “The Analytic Hierarchy Process”, McGraw-Hill Co.:New York, NY, 1980.