A Multiagent System For the Support of Concurrent Engineering

Hu Yongtong, Liu Ping
Yan Yuhong, Zheng Danian, Ma Changchao, Jürgen Bode, Ren Shouju
Automation Department
Tsinghua University
Beijing, 100084, P.R.China
(Email: bxs-oaa@mail.tsinghua.edu.cn)

ABSTRACT

Concurrent design of a product involves the cooperation of many functions such as manufacturing, logistics, marketing, and sales. It can be represented as a multiagent system with each agent representing a kind of expertise. Each agent has its own evaluation criteria and egoistic goals. Decision-making leads to problems typical for Concurrent Engineering, in which all related processes are simultaneously taken into consideration. In this paper, we discuss some characteristics of conflict, conflict type identification as well as conflict resolution. A system architecture and an agent structure are proposed. We also prompt a novel schema named Distributed Dynamic Conceptual Tree (DDCT) to overcome the difficulty of understanding between agents during the problem solving process and reduce communication amount between agents. An elaborated demonstration validating our work is presented.

1. INTRODUCTION

Concurrent Engineering is a framework of methods in which the processes related to the design of a product are taken into consideration simultaneously, ensuring required matching of the product's structural with functional requirements and the associated implications[1].

Many corporate functions such as manufacturing, logistics, marketing, and sales are involved. A lack of communication between all these aspects often lead to unfeasible or sub-optimal designs. In this paper we focus on an integrated system named MULACE (MULtiAgent system for Concurrent Engineering) which is based on a multiagent architecture for the support of Concurrent Engineering. It provides a distributed problem-solving environment and especially a conflict identification and conflict solving schema. Thus, it fosters communication and negotiation between various participants (agents) involved in product design, combines their efforts to achieve a global optimized goal and makes rapid reaction to market changes possible.

A well integrated multiagent system should run with both flexibility and efficiency. Flexibility means that the system can add or remove agents easily, handle exceptions and solve conflicts. Efficiency means the system makes decisions quickly and does not waste system resources. Striking a balance between agent flexibility and efficiency is one of the most prominent problems of the system architecture design[2].

Our integrated environment serves the following important needs:

- System framework: MULACE not only supports individual tasks but also helps them to cooperate to achieve a global optimized solution.
- Public Language: Different agent has different private language sets. When agents' combined efforts are needed to achieve a goal, an interpreter is needed. Then, during communication, agents can assure they will not deviate from the topic.
- Conflict classification and resolution: Agents cooperate and compete with each other, but agents are egocentric at the same time. They possess only a local optimized view for a given task. We establish a novel schema to identify conflict types according to their character, then take different actions to solve them, thus simplifying the conflict resolution process.
- User model: Users can easily participate in the problem solving process through Personal Assistants (PA) [3] agents. One's PA maintains a personal model that includes where one currently is, how one can be reached, what one's preferences are, what information one would like monitored through the knowledge service, and what action to take. PA also interface between real-world experts and expert agents to update system's knowledge.

2. MOTIVATION

Existing research has primarily been concerned with engineering problems in design [3][4][5][6][7]. But to design, produce or sell a high quality product depends on interdependent decisions from almost every department of the enterprise. In this paper, we apply a multiagent system to solve problems encountered in the whole product life cycle, such as the efficiency and flexibility of the system, the language consistency among different units of the enterprise and conflict identification and resolution. This system supports the development of products according to pre-defined goals. These goals could be lower life cycle cost, shorter delivery time, etc.
Here we propose a system and agent structure as well as specific communication scheme between agents. A detailed computational example is elaborated to validate out research.

IEEE International Conference on System, Man and Cybernetics, Oct. 14-17, 1996, Beijing, China

3. MULTIAGENT ARCHITECTURE

3.1 Goal and Properties

Flexibility and efficiency should be taken into consideration simultaneously. In this section, the structure of MULACE is described. It is a compromise between these two important goals. In MULACE, each agent has its own system information module and bulletin board, while there still exists a shared system blackboard. No central management agent controls the decision making process, and all agents are equal from the system structure point of view.

The agents' internal structure and system architecture are described in detail below.

3.2 Agent Structure

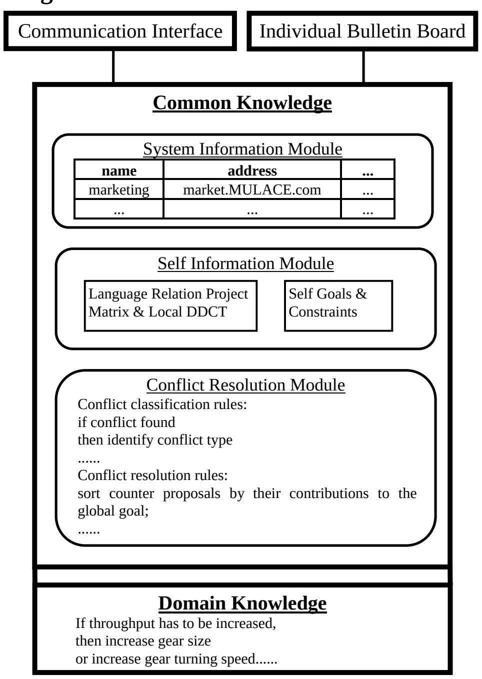

Fig. 1. Structure of Design Agent

Technically, each agent is a specialized expert system, which solves a clearly described problem either by itself or with the help of other agents. The desirable qualities of an agent are:

- Autonomy: An agent is goal-oriented and takes the responsibility of how to satisfy a task.
- Adaptability: An agent can automatically adapt to the environment, which includes the update of the public language, the introduction of new agents as well as knowledge enrichment in existing agents.
- Communication ability: An agent can engage in complex communication with other agents, including human beings to obtain information, call for help to accomplish its goals, negotiate for a compromise or solve conflicts.
- Cooperation: Agents can cooperate to achieve a global goal. An agent can negotiate with other agents, convince other agents, or relax its own constraints when necessary.

In order to accomplish all these features, we present the agent structure depicted in Fig. 1.

Each agent has a private part and a public part. The Private part includes two layers. The first layer contains the agent's domain knowledge base. The second layer contains the knowledge common to all agents which helps agents to interact with the system. There are 3 components in the common knowledge base layer which are described in more detail in the following sections:

- System Information Module: It is used to learn the system information during interaction with other agents, remember the names, addresses, ability and performances of often contacted agents, and forget seldom used one.
- Self Information Module: It is used to help maintain the public language interpreter when new agents are added into the system, to keep the map between private and public language, and to identify conflicts. This Module also contains the goals and constrains of the agent itself.
- Conflict Resolution Module: It includes conflict classification rules and conflict resolution rules, which are used to solve conflicts with other agents.

The public part consist of an individual bulletin board and the communication interface.

3.3 System Structure and Message Passing

The structure of the system influences its performance greatly. There are two communication methods in MULACE: agents can communicate with each other directly, or through the central blackboard.

Using a central blackboard is less efficient but flexible. When an agent needs help, it can simply post information onto the central blackboard which is accessible to all agents. When a new agent is added into the system, it just needs to keep watching the central blackboard and then take part in the decision making process. Thus, inserting new agents requires only little modification to the system.

However, the central blackboard is a bottleneck of communication. A method to solve this problem is to introduce a System Information Module and an Individual Bulletin Board to each agent. The System Information Module records the addresses and performances of often contacted agents in the system (like an extended address book). When help is needed, the agent checks its System Information Module first, trying to send messages directly. Only when the answer is not satisfying, the agent posts help to the central blackboard and broadcasts to all agents. If the agent receives a satisfying answer in this case, it will update its address book noting responding agents' performance in the System Information Module.

At the same time, each agent has an individual bulletin board which is used to gather answers from other agents. Local negotiations and conflict resolution processes are based on this individual bulletin board rather than on the central blackboard. These methods reduce the flexibility to some extent, but save a lot of communication cost.

The agents interact with each other by passing messages. A message consists of 6 parts: 1) message type, 2) message sender, 3) message receiver, 4) work flow number, 5) message body, 6) goal and global constrains.

There are 4 types of interactions: INFORM, REQUEST, INQUIRY and NEGOTIATION. INFORM is intended for passing notifications; REQUEST is used to issue commands or request actions; INQUIRY is designed to acquire information; NEGOTIATION helps to collect other agents' opinions about one agent's partial problem solution.

3.4 Distributed Dynamic Concept Tree System

Another problem is how to make agents understand each other well during the message passing processes. It is often assumed that all agents share a public language and can find out accurately which topic they are talking about. However, in a real environment experts usually have their own private language set. If this is to be realized in a computer-based multiagent system, it is necessary to introduce a method to promote the communication efficiency and keep the system flexibility as well.

3.4.1 DDCT Language interpretation System: The Distributed Dynamic Concept Tree system works as a language interpreter. Its concept tree leaves represent all the concepts of public language set and can be updated dynamically and interactively. It makes conflict detecting and identifying easier and keeps the flexibility at the same time.

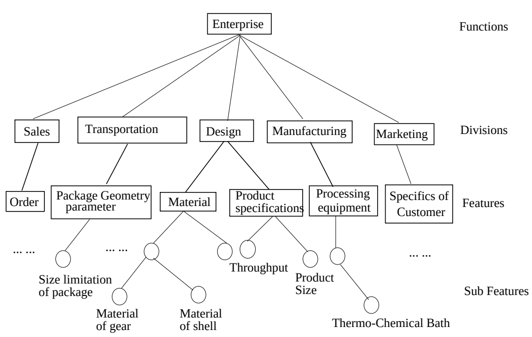

Fig. 2 Structure of Distributed Dynamic Concept Tree

The DDCT System consists of a Concept Tree and Language Relation Project Matrixes and is distributed among all the agents of the system. The Concept Tree is a concept directory built as a tree (Fig. 2).

The Language Relation Project Matrix represents the relationships between the private language set and the DDCT leave concepts (Fig. 3).

The value of the relation factors shows that the related elements of the language sets are corelated.

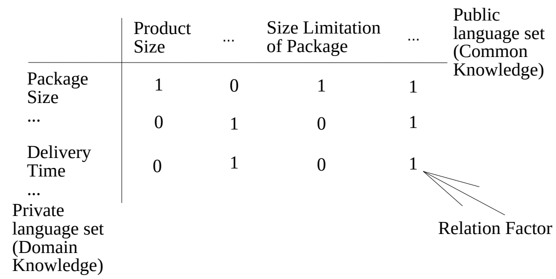

Fig. 3 Language Relation Project Matrix

3.4.2 Usage of DDCT: Before a message is transmitted to an agents or posted onto Blackboard, the key concept of the topic will be added into the Key Concept field in the message body. The receiving agent will check the key concepts in order to find out if the topic of the message is of interest to him.

For example, the transportation agent checks the message from the design agent on the Central Blackboard. In the Key Concept field it finds the words 'Product Size'. In its Language Relation Project Matrix the relation factor between public concept 'Product Size' and its private concept 'Package Geometry Size' is 1. Thus, transportation agent knows that 'Product Size' is closely related to 'Package Geometry Size'.

3.5 Conflict Resolution

3.5.1 Conflict Definition and Classification: The relationship between agents can be classified into four types according to their interactions during the problem-solving process (Table 1). Agents can act in both cooperative and competitive manner[6]. Agents cooperate by working together toward a global goal, and compete by maximally improving their own position. Conflicts exist in competitive inquiry, competitive negotiation, as well as cooperative negotiation.

	Cooperation	Competition
Inquiry	(1)	(2)
Negotiation	(3)	(4)

Table 1. Relationship between Agents

(1) Cooperative inquiry: One agent needs data contributed by another agent. Thus, the latter agent supports the former one.
(2) Competitive inquiry: Two or more agents are eligible to contribute to one inquiry and only one response can be selected.
(3) Cooperative negotiation: Two or more agents' requirements must be satisfied during the negotiation process.
(4) Competitive negotiation: Two or more agents are eligible to one negotiation topic and only one will be accepted.

3.5.2 Conflict Identification: In the communication process, each agent has the responsibility of detecting whether there is conflict, and of identifying its type. If agents have different answers to one inquiry, or have different opinions in negotiation, a conflict occurs. If the key concepts of answers or opinions are different, the conflict is a cooperative conflict, otherwise it is competitive.

For example, the process planning agent and the industrial engineering agent discuss the material of a new pump gear and its processing method. If process planning agent disagrees with industrial engineering agent (e.g. because the existing machinary would not be able to process the new material although it would be less expensive), there exists conflict. Because the key concept of them are both “material of gears”, the conflict is competitive.

3.5.3 Conflict Solving: The problem solving process is initiated by input with specification and constraints. During the negotiation process, agents on the one hand attempt to maximally improve the satisfaction degree of local goals, while on the other hand they are willing to compromise in order to achieve the global goal. All agents should evaluate their contribution to the global goal, such as delivery time or total cost, which should be understood by all agents. Thus, the agents can find out in which direction to compromise and how to trade off between local goals.

The conflict solving process in negotiation can be described as follows:
Step 1: One agent submits a suggestion and related constraints onto the blackboard to initiate the negotiation process;
Step 2: All agents read the negotiation message and send their opinions (agree, disagree, or indifferent) back to the agent who sent the message. Disagreements should be accompanied by a counter proposal and a reason. These replies will appear on the individual board of the message sender.
Step 3: Message sender reads its own bulletin board and evaluates the suggestions. If conflicts exist, it should check the Key Concept to determine the conflict type.
Step 4: If the key concepts of the conflicted idea of two or more agents are the same, a competitive conflict occurs; The conflicted agents should discuss with each other and only the best will be elected to participate the cooperative discussion (competitive conflict resolution).
Step 5: Agent sorts the suggestions from other agents based on their contribution to the global goal, then revises its suggestion following the best proposal (cooperative conflict resolution).
Step 6: Re-posts the new suggestion and go to Step 2.

Through this way, the original message sender deals with conflicts of both cooperative and competitive types within the related agent group.

4. AN ILLUSTRATIVE DEMO: MULACE

The scenario of the demonstration system MULACE is to develop a new gear pump. Eight agents representing large parts of the product life cycle are included in the system. They are design, manufacturing, process planning, industrial engineering, marketing, transportation, sales, and finance. Every agent in the system contributes its knowledge and expertise for the new product plan. The goal is to finally meet all agents’ original constraints or relaxed constraints. Table 2 and Fig. 4 illustrate the problem solving process.

In the current case, the cooperative design procedure is started by the sales agent who presents a customer demand for a gear pump with larger throughput.

The sales agent puts an inquiry message on the Central Blackboard (CBB) (Step 1). Every agent checks the Central Blackboard to find problems it might be able to contribute to.

The design agent checks the message, and gets the product requirements and global goals. So it knows a new product with certain specification is to be designed. The design agent decides to dedicate itself, for the task is in its domain (Step 2). It informs the sales agent that its request is being processed(Not shown in Fig. 4). The design agent uses its domain knowledge to check the existing plans, modifies them to meet the new specifications and gets a new solution, which is to enlarge gear size. Then it posts it onto the Central Blackboard (Step 3).

The other agents read the product plan on the Central Blackboard (Step 4). All the other agents interested in the product plan are invoked. They send their opinions to the Individual Bulletin Board (IBB) of the design agent, for it raised the negotiation.

All agents except the transportation agent and the marketing agent agree with the plan, so they send the agreement messages to the design agent. The design agent then records the addresses and other information of those agents into its System Information Module(Step 5).

The transportation agent considers the total size is too large for the cargo, so it gives a counter proposal on the Individual Bulletin Board of the design agent with the suggestion to reduce product size, and the effect of this counter proposal to transportation cost.

The marketing agent considers the new gear pump is too heavy for the customer for proper handling, so it gives its counter proposal to reduce weight. These conflicts are both cooperative conflicts as their Key Concepts are different (Table 1). So all of those violated constraints should be met simultaneously.

The design agent detects the conflicts and uses its Conflict Resolution Module to solve them. It sorts the contributions of marketing agent and transportation agent, decides to reduce the product size first. After consideration, the design agent

Step	Sender	Work Flow No	Receiver	Body	Goal	Type
1	sales	1	CBB	product possibility; product specification: : throughput>=1,000	cost<$10,000 delivery time<3 Mon.	Inquiry
3	design	1.1	CBB	product plan 1:	as above	Negotiation

				:product specification : housing size= 27 : weight =33 : material = Steel No. 45 : product Size :height =15 :width = 28 :length =31_ Key Concept: product size, product weight... ...		
5	finance	1.1	IBB of design	opinion: indifferent		Inform
5	manufacturing	1.1	IBB of design	opinion: agree Key Concept: material of gear		Inform
5	transportation	1.1	IBB of design	opinion: disagree counter proposal: : height <10; : width<24. contribution to goal: cost saved: $200. Key Concept: package size in transportation.	as above	Negotiation
5	marketing	1.1	IBB of design	opinion: disagree counter proposal: weight < 30 contribution to goal: cost saved: $100. Key Concept: product weight.	as above	Negotiation
6	design	1.2	CBB	product plan 2: :product specification: : housing size= 21; : weight = 20; : material = alloyed steel. : product Size :height =10 :width = 20; :length =25; Key Concept: product size... ...		Negotiation
8	process planning	1.2	IBB of design	opinion: disagree counter proposal: material = Steel No. 45 contribution to goal: cost saved : $300 Key Concept: material of gear .	as above	Negotiation
8	industrial engineering	1.2	IBB of design	opinion: agree proposal: new processing method Key Concept: material of gear.	as above	Negotiation
9	design	1.3	finance	evaluate the two methods process 1: using old machine; process 2: using new process method.	time-limit: 2 days for reply	Request
10	finance	1.3	IBB of design	result of inquiry: process 1: Internal Rate of Return=6.4% process 1: Internal Rate of Return=8.1% Key Concept: product cost		Inform
11	design	1.2	process planning.	inform: denial of suggested process.		Inform
12	design	1.2	industrial engineering	inform: acceptance of suggested process.		Inform
13	design	1	sales	inform: inquiry satisfied.		Inform

Table 2. Decision Making Process of the Demo system
(less important steps omitted; CBB: Central Blackboard; IBB: Individual Bulletin Board; variable units omitted;)

decides to present an alternative plan which adopts the method of increasing the turning speed of the pump instead of gear size. In this case, a new material is needed for the housing which tolerates higher pressure.It presents the new plan on the Central Blackboard (Step 6). At the same time, the design agent informs the transportation agent and the marketing agent that their proposals are accepted. The alternative plan activates another round of negotiation.

All the agents read the revised plan from Central Blackboard (Step 7). This time the transportation agent and the marketing agent both agree with it.

The process planning agent expresses disagreement because it is too expensive to process the new material on the existing equipment. But the industrial engineering agent considers using a processing method that would require new machines (Step 8).

The design agent analyses the messages it received. According to the Key Concept it finds out that two agents want to solve the same problem using different methods. So the conflicts are competitive conflicts (Table 1). Design should evaluate thetwo suggestions and select the least expensive one, but cost estimation is out of its domain. Therefore the design agent sends a request message directly to the finance agent (Step 9).

The finance agent calculates investment cost, and gives the answer back to the design agent. The result is that suggestion the industrial engineering agent is less costly (Step 10) .

The design agent informs the industrial engineering agent and the process planning agent about the final decision (Steps 11,12).

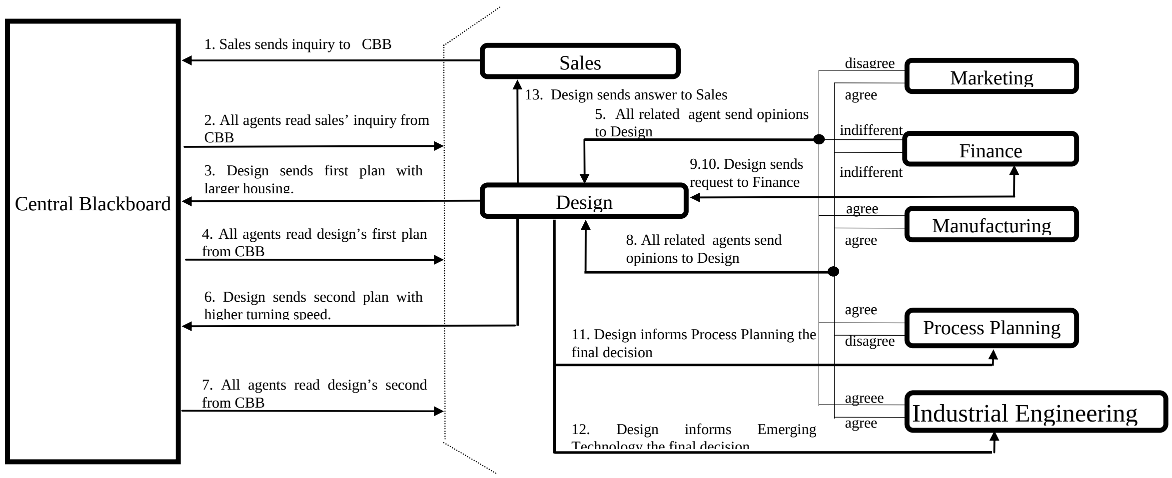

Fig. 4. Decision Making Process of the Demo System

Up to now, all the agents in the community agree to the product plan, and the product specification is also satisfied. A feasible and globally optimal plan is achieved. The design agent informs the sales agent about the result (Step 13).

During the example mentioned above, most conflict types depicted in Table 1 have been elaborated: competitive negotiation in step 8, cooperative negotiation in step 4, and cooperative inquiry in step 10.

5. CONCLUSION

This paper focuses on the application of Multiagent Systems in Concurrent Engineering. The major obstacles are discussed, including efficiency and flexibility of the system, language consistency among agents and identification of conflicts. Methods for the solution of these problems are presented, such as a distributed system architecture, a language interpretation method using Distributed Dynamic Concept tree and a conflict resolution mechanism based on negotiation. The concept design of a computer software system MULACE is presented on a manufacturing enterprise background in order to validate the approach.

Further research will be conducted in the aspects of conflict identification and resolution methods, as well as the improvement of the system in the application environment.

6. REFERENCES

[1] Marc H. Van Liedekerke, "Debugging multi-agent systems", Information and SoftwareTechnology, Vol. 37, No. 2, pp. 103-112.

[2] D. E. Mahling and N. Craven, W. B. Croft, "From Office Automation to Intelligent Workflow Systems", IEEE Expert, Vol. 10, No. 3, June, 1995, pp. 41-47.

[3] Jeff. Y.C. Pan and Jay M. Tenenbaum, "An Intelligent Agent Framework for Enterprise Integration", IEEE Trans. on System, Man, and Cybernetics, Vol. 21, No. 6, November/December, 1991, pp. 1391-1403.

[4] Katia P. Sycara, "Cooperative Negotiation in Concurrent Engineering Design", Computer-Aided Cooperative Product Development, D. Sriram, R. Logcher, S. Fukuda (Eds.), Springer-Verlag, 1991, pp. 269-296.

[5] Hisup Park, "An agent-based approach to concurrent cable harness design", Artificial Intelligence for Engineering Design, Analysis, and Manufacturing , 8, 1994, pp. 45-61.

[6] Keith J, Werkman, Marcello Barone, Stephanie J. Wagaman, John Lsman Wilson, "Evaluating Alternate Connection Designs Through Multiagent Negotiation, Designer Fabricator Interpreter System", Computer-Aided Cooperative Product Development, D. Sriram, R. Logcher, S. Fukuda (Eds.), Springer-Verlag, 1991,pp. 298-329.

[7] Mark Klein, "Supporting Conflict Resolution in Cooperative Design Systems", IEEE Trans. on System, Man, and Cybernetics, Vol. 21, No. 6, November/December, 1991, pp. 1379-1390.