Multi-Agent based Partnering Mechanism for Virtual Enterprise

Toshiya Kaihara
Graduate School of Science and Technology, Kobe University
Kobe, Hyogo 657-8501, Japan

and

Susumu Fujii
Faculty of Engineering, Kobe University
Kobe, Hyogo 657-8501, Japan

ABSTRACT

Nowadays, Virtual Enterprise (VE) is a crucial paradigm of business management in agile environment. In this paper, we focus on negotiation process in VE formulation as a basic research to clarify its effective management. Each enterprise in VE is defined as agent with multi-utilities and a framework of multi-agent programming with game theoretic approach is newly proposed as negotiation algorithm amongst the agents. We develop a computer simulation model to form VE through multiple negotiations amongst several potential members in the negotiation domain, and finally clarify the formulation dynamism with the negotiation process.

Keywords: Virtual Enterprise, Negotiation, Multi-Agent, Game theory, Partnering.

1. INTRODUCTION

Nowadays, Virtual Enterprise (VE) is a crucial paradigm of business management in agile environment. VE exists in both service and manufacturing organizations, although the complexity of the each enterprise in VE may vary greatly from industry to industry. Realistic VE handles multiple end products with shared components, facilities and capacities [1]. Since the flow of materials in VE is not always along an arborescent network, various modes of transportation may be considered, and the bill of materials for the end items may be both deep and large.

Traditionally, marketing, distribution, planning, manufacturing, and the purchasing organizations operated independently. These organizations have their own objectives and these are often conflicting. Marketing's objective of high customer service and maximum sales conflict with manufacturing and distribution goals. Many manufacturing operations are designed to maximize throughput and lower costs with little consideration for the impact on inventory levels and distribution capabilities. Purchasing contracts are often negotiated with very little information beyond historical buying patterns. The result of these factors is that there is not a single, integrated plan for the organization - there were as many plans as businesses. Clearly, there is a need for a mechanism through which these different functions can be integrated together. Although cooperation is the fundamental characteristic of VE concept, due to its distributed environment and the autonomous and heterogeneous nature of the VE members, cooperation can only be succeed if a proper management of dependencies between activities is in place just like Supply Chain Management [2] [3].

In this paper, we focus on negotiation process in VE formulation as a basic research to clarify its effective management. Each enterprise in VE is defined as agent with multi-utilities and a framework of multi-agent programming with game theoretic approach [4] is newly proposed as negotiation algorithm amongst the agents. Each unit is defined as agent in our VE model, and their decision makings are formulated as a game theoretic methodology. We adopt CNP (Contract Net Protocol) [5] [6] as the coordination and negotiation mechanism amongst the units. CNP models transfer of control in a distributed system with the metaphor of negotiation among autonomous intelligent beings. CNP consists of a set of nodes that negotiate with one another through a set of message [7] [8] [9]. Nodes generally represent the distributed computing resources to be managed, correspond to "enterprises" in this paper. We develop a computer simulation model to form VE through multiple negotiations amongst several potential members in the negotiation domain, and finally clarify the formulation dynamism with the negotiation process.

There is a limit of 12 single spaced, double-column pages for each article in the Journal. At most 8 additional pages can be included, as long as a fee of US$ 100.00 per extra page is covered by the author. Regular papers must not exceed 20 pages, in any case. Be sure to fill out and send with your camera-ready paper, your JSCI Copyright Transfer Form.

The deadline for receiving your camera-ready paper is three weeks after the paper's acceptation for publication. Both, an electronic copy, in PDF, and a hard copy of the final paper should be sent. The electronic copy should be sent via web page and/or via e-mail attachment, and the hard copy should be sent to the following address:

2. ENTERPRISE AGENT

Virtual enterprise model

Virtual enterprises are defined as "agile" enterprises, i.e. as enterprises with integration and reconfiguration capability in useful time, integrated from independent enterprises, primitive or complex, with the aim of taking profit from a specific market opportunity. After the conclusion of that opportunity, the virtual enterprise dissolves and a new virtual enterprise is integrated, or it reconfigures itself in order to achieve the necessary competitiveness to respond to another market opportunity. General definition of VE is as follows [1]:

> "A virtual enterprise is a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities and whose cooperation is supported by computer networks."

For the independent, primitive or complex enterprises, or companies, candidates to integrate a V E, we will use the designation "resource", the sense that from the point of view of the virtual enterprise, these enterprises represent the potential "resources" for integration. It is important to notice that the resource is a recursive construct; resources can be primitive or complex.

The knowledge and physical resources associated to the development and production of most of today's products often exceed what a single firm is able to accomplish. The new production enterprise is a network that shares experience, knowledge and capabilities – it is critical in this new environment for a manufacturing company to be able to efficiently tap these knowledge and information networks.

The organisational challenge of partitioning tasks among partners in the distributed manufacturing environment so that they fit and take advantage of the different competencies in V E, integration of the same, coordination and reconfigurability in order to keep alignment with the market requirements, is of main concern, and can determine the success or failure of a project.

Face to the requirements of competitiveness that the competition environment is demanding, enterprises are expected to present at least the following characteristics:

- Fast reconfigurability or adaptability: the ability of fast change face to the unpredictable changes in the environment / market, implying substitution of resources (transition to a new A/V E instantiation)
- Evolutionary capability: the ability to learn with history.

A large number of diversified networked organisations of enterprises fall under the general definition of VE. We assumed our VE model in the possible simplest definition as a basic research, as follows:

i) Duration: Single business
An alliance of the enterprises is established towards a single business opportunity, and is dissolved at the end of such process.

ii) Topology: Fixed structure
There exist established supply chains with an almost fixed structure.

iii) Participation: Single alliance
All the enterprises are participating into only a single alliance at the same time.

iv) Coordination: Democratic alliance
A different organisation can be found in some supply chains without a dominant company. All the enterprises cooperate on an equal basis, preserving their autonomy.

v) Visibility scope: Single level
All the enterprises in VE communicate only to its direct neighbours in its architecture (figure 1). That is the case observed in most supply chains.

Needless to say, it is very important and difficult activity in forming a virtual enterprise to select appropriate business partners, i.e. partnering, because each enterprise considers not only pursuing its profit but also sharing the risk to join the virtual enterprise. The partnering is described as coordination activity amongst the enterprises, and some sophisticated coordination mechanism is required to realise efficient interactions.

The development of coordination mechanism in computer science can be found in the area of workflow management system, computer supported cooperative work (CSCW), and multi-agent systems.

The area of multi-agent systems, especially when involving intelligent autonomous agents, has been discussing coordination issues and supporting mechanism [7]. The interaction capability, both amongst agents and between agents and their environment, is one of the basic characteristics of an agent.

In this paper we focus on the contract net protocol (CNP), that is one of the mechanism coming from the early works on multi-agent systems [5], as the coordination and negotiation mechanism amongst business units in VE.

Figure 1 shows the assumed VE model in this paper. We call an enterprise as unit, and there exist m layers, which have mn units in the VE model. The lowest level corresponds to consumers who can create original task requests to the VE. As the layer number, m, increases, we describe it 'lower' based on the product flow order in this paper.

At first, the customer dispatches new order to all the units in layer m, and then several units, which are satisfied with the order, responds and circulates the order toward upper units in the VE model. Finally a VE with single supply chain will be established for the order as a consequence of their negotiations through all the layers.

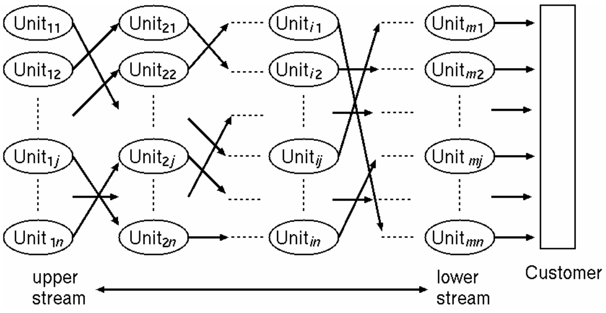

Figure 1 VE model

Unit structure

Each unit is defined as agent in our VE model, and its structure is described in figure 2. We adopt CNP as the coordination and negotiation mechanism amongst the units. CNP models transfer of control in a distributed system with the metaphor of negotiation among autonomous intelligent beings. CNP consists of a set of nodes that negotiate with one another through a set of message. Nodes generally represent the distributed computing resources to be managed, correspond to "units" in this paper.

An agent (=unit) can act both as a manager and a contractor of a delivery sets. When a unit receives new order (= task announcement) *i*, it creates a contractor / manager set (Manager *i* / Contractor *i*) for the task inside. Manager *i* creates a new order towards the lower units to secure the contract with the upper layer.

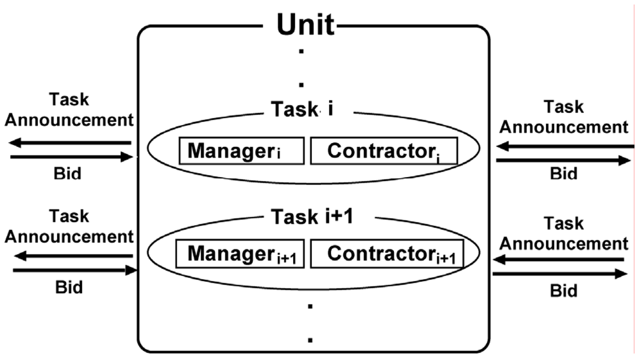

Figure 2 Unit structure

Basic assumption

There exist several situations in partnering amongst enterprise agents. In this paper it is assumed that the product demand is predictable in the negotiation under multipurpose criterion. That means order patterns are previously given and the negotiations start after the order reached to each enterprise agent. They should prepare robust solutions with maximum utilities against the order. We propose agent behaviours based on game theoretic approach according to this assumption.

Negotiation algorithm

The timeline of the proposed negotiation mechanism in this paper is shown in figure 3. Negotiation steps according to agent roles are described as follows:

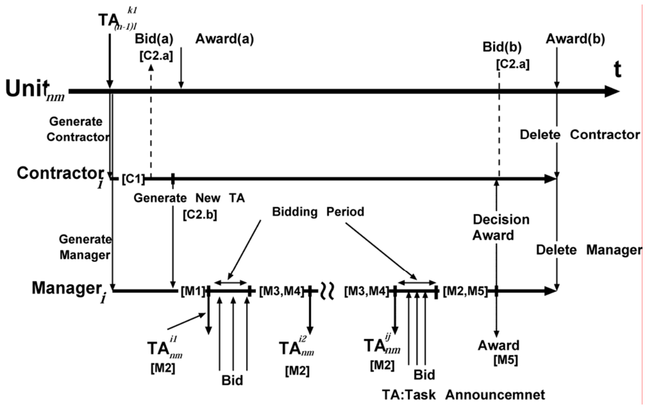

Figure 3 Negotiation flow

Manager

Step M1: Create a new task based on the received bid information.

Step M2: Task announcement (TA) to the lower units.

Step M3: After the bidding period expired, check all the acquired bids according to its standard. If there exists no bid to select, go to M4. Otherwise go to M5.

Step M4: Modify the task and go to M2.

Step M5: Select the task and send reward (Reward) to the corresponding unit.

Contractor

Step C1: Create an estimated bid.

Step C2: Send the bid.

Step C3: Request task announcement to the manager.

Agent behaviour

In this model, all the orders are clearly given before the negotiations. Agent behaviours are described in each negotiation step.

- Bidding (Step C2)

Each contractor (U_{ij}) has three attributes, such as cost, lead time and quality, in their bid for order k defined as follows:

$$Cost\,_{ij}^{k} = E_{ij}^{k} + D_{ij}^{k} + P_{ij}^{k} \tag{1}$$

$$Leadtime\,_{ij}^{k} = \lambda_{ij}^{k} / E_{ij}^{k} \tag{2}$$

$$Quality\,_{ij}^{k} = \mu_{ij}^{k} D_{ij}^{k} (1 - \exp^{-v_{ij}^{k} P_{ij}^{k}}) \tag{3}$$

where

$Cost\,_{ij}^{k}$: total cost for U_{ij} to process order k

$Leadtime\,_{ij}^{k}$: lead time for U_{ij} to process order k

$Quality\,_{ij}^{k}$: product quality for U_{ij} to process order k

$E_{ij}^{k}, D_{ij}^{k}, P_{ij}^{k}$: equipment / development / personnel cost

λ_{ij}^{k} : coefficient of leadtime

μ_{ij}^{k}, v_{ij}^{k} : coefficients of quality

Cost vs. lead time, and cost vs. quality, are in trade off relationship in those equations with reality.

- Reward (Step M5)

After the bidding by contractors, managers compute following pay-off matrix according to their utilities against all the bids.

$$\begin{array}{c} \\ Unit_{i1} \\ Unit_{i2} \\ \cdots \\ Unit_{ij} \\ \cdots \\ Unit_{in} \end{array} \begin{array}{c} \begin{array}{ccc} C & L & Q \end{array} \\ \begin{bmatrix} C_{i1}^{k} & L_{i1}^{k} & Q_{i1}^{k} \\ C_{i2}^{k} & L_{i2}^{k} & Q_{i2}^{k} \\ \cdots & \cdots & \cdots \\ C_{ij}^{k} & L_{ij}^{k} & Q_{ij}^{k} \\ \cdots & \cdots & \cdots \\ C_{in}^{k} & L_{im}^{k} & Q_{im}^{k} \end{bmatrix} \end{array} \tag{4}$$

where

$$C^k_{ij} = (\overline{c^k_i} - Cost^k_{ij}) / s_{c^k_i} \tag{5}$$

$$L^k_{ij} = (\overline{l^k_i} - Leadtime^k_{ij}) / s_{l^k_i} \tag{6}$$

$$Q^k_{ij} = (Quality^k_{ij} - \overline{q^k_i}) / s_{q^k_i} \tag{7}$$

C^k_{ij} : utility on cost for U_{ij} to process order k

L^k_{ij} : utility on lead time for U_{ij} to process order k

Q^k_{ij} : utility on quality for U_{ij} to process order k

$\overline{c^k_i}, \overline{l^k_i}, \overline{q^k_i}$: average utility of all the bid on cost, lead time, quality

$s_{c^k_i}, s_{l^k_i}, s_{q^k_i}$: standard deviation of all the bid on cost, lead time, quality

Five strategies are defined as selection mechanism using the payoff matrix in (4).

- method 1: cost minimisation strategy

$$\max_{j=1,2,\dots,n} = C^k_{ij} \tag{8}$$

- method 2: lead time minimisation strategy

$$\max_{j=1,2,\dots,n} = L^k_{ij} \tag{9}$$

- method 3: quality maximisation strategy

$$\max_{j=1,2,\dots,n} = Q^k_{ij} \tag{10}$$

- method 4: total utility maximisation strategy

$$\max_{j=1,2,\dots,n} = C^k_{ij} + L^k_{ij} + Q^k_{ij} \tag{11}$$

- method 5: max-min strategy

$$\max_{j=1,2,\dots,n} \min\{ C^k_{ij}, L^k_{ij}, Q^k_{ij} \} \tag{12}$$

Method 5 is so called " max-min strategy" in game theory, which has been proved to conduct Nash equilibrium solution by min-max theorem, if the game is in zero-sum situation like our formulation shown in equation (4).

Reinforcement learning

In non-sequential tasks, a unit must learn a mapping of situations to actions that maximises the expected immediate payoff in general. Putting in a better context, units have their actions-to-reward manual by their side. Sequential tasks are more difficult because actions selected by units may influence its future situations and thus its future payoffs. In this case, the unit interacts with the environment over an extended period of time, and it needs to evaluate its actions on the basis of their long-term consequences. This involves a credit assignment problem, i.e. a whole sequence of actions takes place before long term consequences are known. This would be difficult because actions in a sequence may have different values with respect to the consequences.

Q-learning is a recent form of reinforcement learning algorithm that does not need a model of its environment and can be used on-line. Therefore, it is very suited for repeated games against an unknown opponent. Q-learning algorithm works by estimating the values of state-action pairs. The value $Q(s,a)$ is defined to be the expected discounted sum of future payoffs obtained by taking action a from state s and following an optimal policy thereafter. Once these values have been learned, the optimal action from any state is the one with the highest Q-value. After being initialised to arbitrary numbers, Q-values are estimated on the basis of experience as follows:

From the current state s, select an action a. This will cause a receipt of an immediate payoff r, and arrival at a next state s'.

We update $Q(s,a)$ based upon this experience as follows:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma \max Q(s',b)) \tag{13}$$

where α is the learning rate and $0 < \gamma < 1$ is the discount factor. This algorithm is guaranteed to converge to the correct Q-values with the probability one if the environment is stationary and depends on the current state and the action taken in it; called Markovian, a lookup table is used to store the Q-values, every state-action pair continues to be visited, and the learning rate is decreased appropriately over time. This exploration strategy does not specify which action to select at each step. Experiments with Q-learning agent have been done in the past with favourable results.

It is quite natural to assume enterprise units have an adaptive behaviour in terms of negotiation process. We especially focus on the bid creation mechanism in step C1, and introduce the reinforcement learning algorithm into it. Contractor's behaviour in each enterprise unit is improved by its experience so as to dispatch a better offer to attain the rewards from managers. The state and the action for Q-Learning, are defined in VE model as follows:

State: State is defined by two kinds of attributes, cost and lead time, because quality is automatically defined by these parameters in (7). State domain is divided by $N*N$ space, and unit recognises its state in the state domain map shown in figure 4.

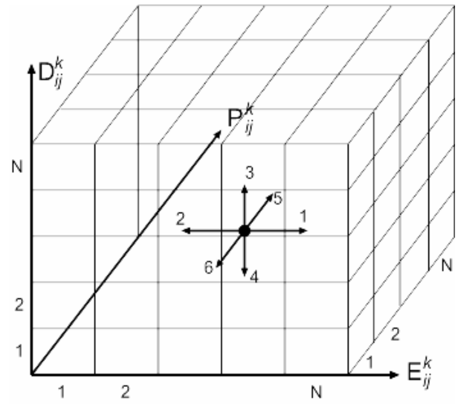

Figure 4 State domain map

Action: Seven types of actions are prepared to respond adaptively to the state as follows:

Action 1: increase E_{ij}^{k}

Action 2: reduce E_{ij}^{k}

Action 3: increase D_{ij}^{k}

Action 4: reduce D_{ij}^{k}

Action 5: increase P_{ij}^{k}

Action 6: reduce P_{ij}^{k}

Action 7: no changes

“Increase” or “reduce” means to move into the divided space next to the current position upwards or downwards in figure 4.

Rewards: Manager unit gives the improved value of each attributes as reward to the negotiating contractor unit.

3. SIMULATION RESULTS

Simulation parameters

A 3-layered VE model for computer simulation was developed to clarify VE formulation dynamism with the proposed negotiation mechanism. Each layer consists of 5 enterprises in this simulation model described in Figure 1.

Simulation parameters are shown in Table 1. All the results are the average of 500 trials in each simulation scenario.

Table 1 Simulation parameters

m	n	N	E	D	P	λ	μ	v
3	5	1	5-15*	5-15*	5-15*	300	5	0.1

* followed by uniform random distribution

Simulation results

Simulation results in terms of negotiation attributes and utilities are shown in Table 2, 3, respectively. All the results are shown in the average (AVE.) and the standard distribution (St. Dev.) in Table 2. Figure 5 also illustrates the average of each negotiation attribute to compare the proposed methods.

Table 2 Simulation results (Negotiation attributes)

	Method 1		Method 2		Method 3		Method 4		Method 5	
	Ave.	St. Dev.	Ave.	St. Dev.	Ave.	St. Dev.	Ave.	St. Dev.	Ave.	St. Dev.
Cost	72.87	5.08	99.89	7.47	104.17	6.61	98.96	8.79	91.19	5.54
LTime	119.53	18.47	68.32	4.84	90.21	18.06	80.72	11.04	91.04	12.24
Quality	65.10	10.58	92.59	18.05	132.34	12.44	106.84	20.22	93.23	12.97

We summarise the characteristics of each method as follows:

Method 1: Cost minimisation

Since the negotiation amongst enterprises is cost-oriented in this method, cost parameter is the best of all the methods in Ave. Additionally Cost and Quality are better in St. Dev., because they are correlated to $E_{ij}^{k}, D_{ij}^{k}, P_{ij}^{k}$ shown in (1) and (3). It has been observed that all the utilities are small to minimise total cost in this method at Table 3.

Table 3 Simulation results (Utilities)

	Method 1	Method 2	Method 3	Method 4	Method 5
E_{ij}^{k}	8.13	13.35	9.96	11.68	10.38
D_{ij}^{k}	8.13	10.11	12.84	11.47	10.61
P_{ij}^{k}	8.03	9.84	11.92	9.83	9.41

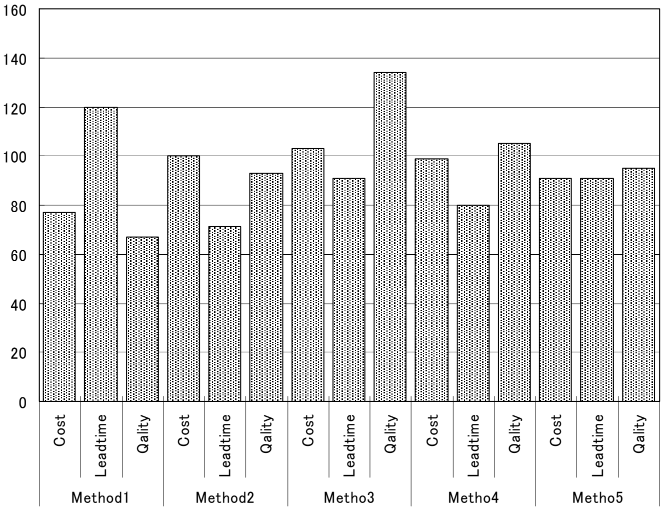

Figure 5 Method comparison

Method 2: Lead time minimisation

Lead time-oriented method naturally conducts the minimal LeadTime in Ave. and St. Dev. Cost parameter is not good, because Leadtime and E_{ij}^{k} is in trade-off relation in (2), and that lead to higher cost.

Method 3: Quality maximisation

It is obvious that quality maximisation strategy caused the worst in Cost, and this result fits well to our general sense. In this method enterprises don’t pay any attention to LeadTime shown in (2).

Method 4: Total utility maximisation

Generally the result is moderate in the balance amongst 3 parameters by trying to maximise total utility. In figure 5 it has been observed that they are relatively better in its

LeadTime and Quality, but worse in its Cost. That is because the general relationship amongst $E_{ij}^{k}, D_{ij}^{k}, P_{ij}^{k}$, in (1), (2) and (3), that means enterprise agents sacrifice LeadTime to increase Cost and Quality.

Method 5: Max-min strategy

Acquired result is completely well-balanced in Ave. It has also been confirmed that this strategy conducts minimal in St. Dev., and their negotiation is stable and robust enough to deal with agile trading situations. That is because each agent tried to minimise the risk based on the min-max theory.

Learning effects

Simulation parameters in Q-Learning algorithm are as follows:

α	: 0.5
γ	: 0.2
the number of tasks for initial learning	: 2000

Simulation results in terms of negotiation attributes and utilities are shown in Table 4. All the results are also shown in the average (AVE.) and the standard distribution (St. Dev.) in Table 4.

Table 4 Q-Learning effects

	Conventional		Q-Learning	
	Ave.	St. Dev.	Ave.	St. Dev.
Cost	1509	124.80	1496	100.60
Lead time	59.48	4.36	59.38	3.76
Quality	298.20	30.62	295.50	24.81

It is obvious that the standard deviation is smaller in Q-Learning based negotiation compared with the conventional negotiations, although there isn't so much difference in the average.

A manager in a unit has to select one contractor amongst several bids at least by relative evaluations in our negotiation mechanism. Therefore it is quite natural that the randomness of the attributes in bid causes large standard deviation in their compromised trade. On the other hand, contractor's bidding strategy based on Q-Learning algorithm produces more acceptable attributes in their bid, and that concludes little randomness with small standard deviation. It is obvious that Q-Learning based bidding strategy leads stable trading between VE units.

Although our investigations have clearly not been exhaustive, it is already apparent that the agent (i.e. unit) behaviours have a great influence on autonomously formulated VE structure as a basic study. Proposed game theoretic formulation on agent decision mechanism with multi-agent paradigm is quite reasonable to realise negotiation process amongst enterprises.

4. CONCLUSIONS

In this paper, we focused on negotiation process in VE formulation as a basic research. Each enterprise in VE was defined as agent with multi-utilities and a framework of multi-agent programming with game theoretic is newly proposed as negotiation algorithm amongst the agents. Each unit is defined as agent in our VE model, and their decision-makings are formulated as game theoretic methodology and Q-Learning algorithm.

Simulation results have proved that the proposed game theoretic formulation on agent decision mechanism with multi-agent paradigm is quite reasonable to analyse negotiation process amongst enterprises.

5. ACKNOWLEDGEMENTS

This research was supported by International Research program IMS (Intelligent Manufacturing System) of IMS centre Japan, under contract No.0219 (HUTOP project), and the research grant-in-aid from Japan Society for the Promotion of Science, contract No. 14580478.

6. REFERENCES

[1] Camarinha-Matos, L. M. et al., The virtual enterprise concept, **Infrastructures for virtual enterprises**, Kluwer academic publishers, Boston, pp.3-14, 1999.

[2] Fisher M. L., Making supply meet demand in uncertain world, **Harvard Business Review**, May/Jun, 1994.

[3] Goldratt E. M., **The GOAL**, North River Press, 1983.

[4] Von Neumann, J. et al., **Theory of Games and Economic Behavior**, Princeton University Press, 1947.

[5] Smith, R., The contract net protocol, **IEEE Transaction on Computers**, C-29, pp.1104-1113, 1980.

[6] Durfee, E. et al., Coherent cooperation among communication problem solvers, **IEEE Transaction on Computers**, N 36, pp.1275-1291, 1987.

[7] Kaihara, T. and S. Fujii, A study on virtual enterprise coalition with multi-agent technology in agile manufacturing environment, **International Journal of Advanced Manufacturing Systems**, Vol.1, No.2, pp.125-139, 2002.

[8] Kaihara, T. and S. Fujii, IT based Virtual Enterprise Coalition Strategy for Agile Manufacturing Environment, **Proc. of the 35th CIRP Int. Seminar on Manufacturing Systems**, pp32-37, 2002.

[9] Kaihara, T. and S. Fujii, A proposal on negotiation methodology in VE, **Collaborative Business Ecosystems and Virtual Enterprises**, Kluwer Academic Publishers, Boston, pp125-132, 2002.