

[論 文]

ソフトウェアエージェントによる価格交渉戦略の比較

A Study on Price Negotiation Strategies Among Software Agents

亀山 裕樹¹, 岡田 ロベルト², 藤井 章博²

Hiroki KAMEYAMA, Roberto OKADA and Akihiro FUJII

¹ 日本アイビーエムビジネスソリューション(株) IBM Global Services Japan Business Solutions Company

² 宮城大学 事業構想学部 デザイン情報学科 School of Project Design, Miyagi University

Abstract

Recent years are witnessing a big increase in the amount of B2B electronic commerce sites, and software agent technology can help. We are exploring different price negotiation strategies to be included in an agent-based B2B electronic market for perishable goods, named GEM (**G**oods-agents in **E**lectronic **M**arkets). In order to experiment on the different possible strategies, we have developed an agent-based price negotiation simulator for GEM, named ATRAS (**A** **TR**ading **A**gents **S**imulator). The negotiating agents set their dealing criteria and start the price negotiation based on the strategy implemented, until the proposed price satisfies both parties. In this paper, we report with focus on the price difference PriceDif - which is the price increment / decrement from the previous offer - against the number of exchanged messages. From our experiments, we found some interesting relations between PriceDif, the success rate and the agreed price.

Key Words : Software Agents, B2B electronic market

キーワード：キーワード：ソフトウェアエージェント, 企業間電子商取引

概 要

近年、企業間電子商取引サイトは増加しており、このようサイトで知的エージェントによる取引支援は注目されておる。本研究では、ソフトウェアエージェントを実装した価格交渉シミュレーター・ATRAS (A TRading Agents Simulator) を開発した。これは現実に行われている電子商取引を想定した様々な交渉戦術のシミュレーションを行うことを目的として開発している。本稿では価格交渉において売り手と買い手が、それぞれ商品を販売する基準となる価格や購入する基準となる価格を保持すると仮定して、その価格から両者の妥協点を探して交渉を続ける場合に注目する。特に交互に取引価格を提示する様々な価格交渉において、前回の金額提示からの譲歩金額 (PriceDif) に注目した。シミュレーションの結果よりメッセージのやりとりとともに PriceDif を変動させていくことで、交渉の成功率を表す Success_Rate と成立時の価格を表す Agreed_Price に一定の規則性があることを確かめることができた。尚、ATRAS で行ったシミュレーション結果に基づいて、今後 ATRAS の交渉プロトコルを B2B システムに実装を行う。

1. Introduction

Recent years are witnessing a big increase in the amount of transactions over the Internet in B2C (Business-to-Consumer) as well as B2B (Business-to-Business). Regarding B2B, the shift of the procurement's market into the electronic commerce can be beneficial for both buyers and sellers, as the buyers can get better lower prices during procurement, while sellers can offer convenient prices.

The authors were involved with the development of an agent-based B2B market for trading of perishable goods such as fresh fish [1]. The introduction of software agent technologies into such electronic commerce market can be beneficial in terms of increase in transaction efficiency when compared with traditional electronic commerce sites. This increase is influenced by better management of good's information and time saving from automated negotiations, performed by software agents.

Thus, it is fundamental to understand the behavior of such software agents in electronic markets. In order to fulfill this need, we have developed ATRAS, which stands for "A TRading Agents Simulator". We then have performed simulation of several negotiation strategies and compared the outcomes. This comparison will help us selecting the strategy more appropriate for the B2B market which we are building.

In the following sections, we first explain the development of ATRAS. Next, we talk

about the different strategies simulated and we close with a summary of our findings.

2. Development of ATRAS (A TRading Agents Simulator)

2.1 Importance of negotiation simulations

We are exploring the information processing functions necessary for software agents to be able to perform price negotiation between Buyer and Seller agents.

Agents are semi-autonomous computer programs which perform some tasks on human's behalf. Regarding agent-mediated electronic markets, the human buyers, sellers and in some cases the intermediaries were modeled as agents. Some examples are Kasbah[2], Tete-a-Tete, MAGMA[3], SICS[4] and works by the IBM Information Economics project[5]. In Kasbah and Tete-a-Tete, users may assign the task of buying or selling a specified good to an agent, which then performs negotiation and settlement of deals, fully automatically, according to the user's choice of predefined strategy.

Our claim is that such models are not enough for real-world B2B applications, so we introduce the concept of modeling the goods themselves as agents [6][7]. We believe that by modeling the goods as agents in B2B electronic markets, the efficiency of processing information regarding a particular good as well as the negotiation process will improve.

By using such software agents to automate price negotiation, a variety of negotiation strategies become possible. Thus, the need for selecting the proper strategy according to the situation is fundamental. In order to help understanding the properties of such negotiation strategies, we have developed the ATRAS simulator, and performed several simulations, taking into account the price as a key factor in the negotiation process.

2.2 Features of ATRAS

ATRAS is a system where a variety of negotiation strategies can be simulated by setting simulation parameters, such as the `Strike_Price` (i.e. the desired maximum price in case of buyer), `make_Offer1` (price offered the first time), `make_Offer2` (price offered the second time) and other conditions for the approval or rejection/breakdown of a particular negotiation.

Different negotiation strategies can be flexibly simulated in ATRAS. An important

function is to provide a place where a large number of dealings between buyers and sellers can be handled simultaneously, so it is possible to analyze statistically the outcomes.

2.3 Development of ATRAS

ATRAS is composed of several modules, such as negotiation module and transaction recording module. This last module is responsible for keeping the record of transaction information into a log file, which can be used later for analysis of the negotiation strategies. The transaction information to be recorded include the `Strike_Price` for both Buyers and Sellers, the messages exchanged, the agreed prices and the conditions hold at the time of success or failure of negotiations.

The development was based on a MarketPlace program written in Java[7], using VisualAge for Java, from IBM.

2.4 Agent types and their behaviors

ATRAS is a simulator which implements the software agent functions necessary for automated negotiations, the multi agent environment for exchange of messages, and negotiation strategies to be studied.

The three types of agents participating in ATRAS and their behavior are described below.

(1) Seller Agent

The agent selling goods in ATRAS is called Seller Agent, or Seller for short.

Each Seller maintains a list of goods to sell, named `Item_List`. Information about the good to sell and the desired selling price `S_Strike_Price` are kept in this list.

The Seller who wants to participate in the market has to register his information as well as the good to sell with the Facilitator Agent, or Facilitator for short. This registered information is used by the Facilitator to do the matchmaking between Buyers and Sellers sharing interest in a particular good.

(2) Buyer Agent

The agent dealing with Seller in order to buy goods is called Buyer Agent, or Buyer for short.

Buyer maintains a list of goods interested in buying, named wishList. Buyer can participate in the market by registering with the Facilitator.

Buyer agent takes an item from its wishList. It then asks the Facilitator to introduce a recommended Seller who has advertised its ability to sell that item.

The major data members include wishList (a vector of items the agent wants to buy and the desired purchase prices), inventory (contains all of the items the Buyer has purchased), and negotiations (a hashtable of negotiations in progress). It goes to sleep for a while, and then if it still has items on its wishList and no negotiations in progress, it takes the first item off the wishList and kicks off a negotiation by asking the Facilitator to recommend a Seller for the item.

(3) Facilitator Agent

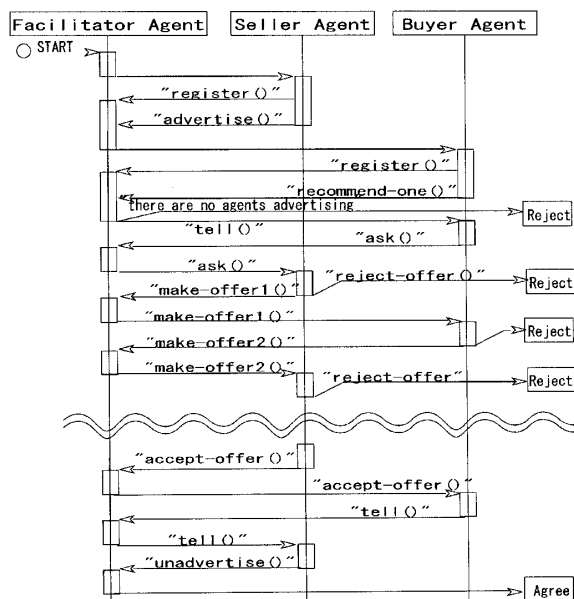
The agent which acts as a mediator/matchmaker between Buyers and Sellers is called Facilitator Agent, or Facilitator for short. The tasks of Facilitator include the matchmaking between Buyers and Sellers sharing interest in buying/selling a particular good, and the post-processing for both Buyer and Seller to leave the market after the dealing ends.

2.5 Message exchange flow in ATRAS

In order to make clear the specification of the system to be developed, we have drawn several UML (Unified Modeling Language) diagrams. One sample is shown in Figure 1, representing the sequence diagram of messages exchanged between the different agents. A more detailed description of the protocol follows.

During the initialization of the dealing process, Seller registers the good to sell with the Facilitator, while Buyer registers the goods he is interested in purchasing.

Fig. 1 : UML sequence diagram of the negotiation protocol in ATRAS



Based on the registered information, Facilitator performs the matchmaking between Buyers and Sellers.

Buyers and Sellers sharing interest in a particular good start the negotiation process by exchanging messages. `Strike_Price` is a variable containing the desired price for Buyers as well as for Sellers, which is used as a base for the calculation of the offered price to the dealing counterpart.

The amount of money of the concession from the last amount of money presentation is called `PriceDif`. The price when the negotiation is approved is called `Agreed_Price`.

If the Buyer and Seller can reach an agreement over a price, the negotiation is successful or approved. On the other hand, if such agreement fails, the negotiation is considered a failure or breakdown. In summary, the following steps are necessary:

1. Buyer asks the Facilitator to recommend-one seller for an item P
2. Facilitator tells the Buyer the name of the Seller
3. Buyer asks the Seller (through the Facilitator) if the Seller has an item P for sale
4. Seller either makes-offer to the Buyer, passing the item P, a unique item id, and an initial asking price, or will deny that he has item P for sale
5. Buyer then can either accept the offer by echoing the offer back to the Seller, or make a counter-offer (with a different price) to the Seller
6. The Seller can either accept the offer, make a counter-offer, or reject the offer
7. If the Seller accepts, he get a `tell` message from the Buyer, and the sales transaction is complete. If the Seller rejects the offer, the negotiation is over.

3 . Price Negotiation Simulations based on ATRAS

Seller is interested in selling the good at the highest possible price, while Buyer is interested in buying at the cheapest possible price. In case of Seller, in case that he cannot agree with the price proposed by the Buyer, he make a counter-proposal by increasing the proposed price. On the other hand, if Buyer cannot agree with the price proposed by Seller, he makes a counter-offer by decreasing the price offered by Seller.

The `Strike_Price` is generated randomly based on a normal distribution probability curve for one of the dealing parts (Buyer or Seller), while the `Strike_Price`

corresponding to the other dealing part is constant.

As the output of simulations, the influence of the different negotiation strategies on the rate of successful or agreed dealings (Success_Rate) and the price at the time of approval (Agreed_Price) are studied.

In this paper, we report on three simulations:

1. Simulation based on One_Shot_Strategy
2. Simulation based on Seller-driven Change_PriceDif_Strategy
3. Simulation based on Buyer-driven Change_PriceDif_Strategy

In the following, we provide the details of the strategies' influence on the price negotiation.

3.1 Simulation of One_Shot_Strategy

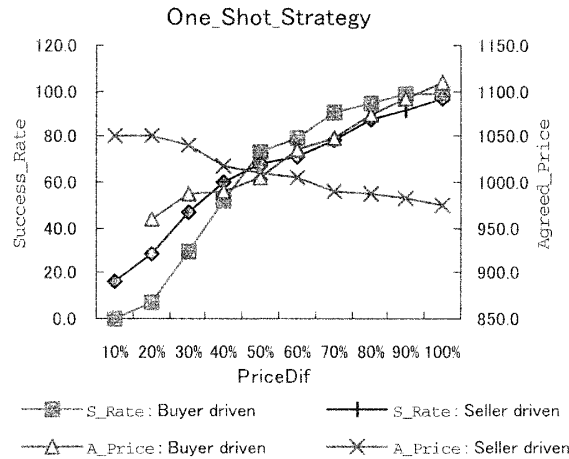
The parameter PriceDif indicate the difference between the new price to be counter-proposed and the price proposed by the other trading part. In One_Shot_Strategy, after receiving an offer from the dealing peer, only one counter-proposal is allowed. If this counter-proposal is accepted, then the negotiaion is considered successful. Otherwise, no further counter-proposal is allowed and the negotiation fails.

Under such conditions, the influence of PriceDif on the Success_Rate and

Table1 Simulation output for One_Shot_Strategy

PriceDif	Buyer-Driven		SellerDriven	
	S_Rate	A_Price	S_Rate	A_Price
10%	0.0		16.2	1050.4
20%	7.6	960.5	28.2	1050.3
30%	29.0	986.6	46.6	1041.1
40%	52.2	991.0	60.2	1017.6
50%	73.0	1006.2	68.6	1010.4
60%	79.8	1036.7	70.8	1005.2
70%	90.4	1048.3	78.2	990.9
80%	94.8	1074.1	87.0	988.3
90%	99.0	1090.8	91.8	981.4
100%	99.0	1110.5	96.2	973.8

Fig.2: Simulation results for One_Shot_Strategy



Agreed_Price is studied. The results are summarized in Table 1 and plotted in Fig. 2, where S_Rate and A_Price represents, Success_Rate and Agreed_Price respectively.

Buyer-driven PriceDif variation:

For the Buyer-driven case i.e. varying the PriceDif of the Buyer in order to calculate a counter-proposal to the price proposed by the Seller, the Success_Rate and Agreed_Price are directly proportional to the PriceDif. With the decrease in PriceDif, Success_Rate decreases notoriously.

Seller-driven PriceDif variation:

For the Seller-driven case i.e. varying the PriceDif of the Seller in order to calculate a counter-proposal to the price proposed by the Buyer, the Success_Rate and Agreed_Price are inversely proportional to the PriceDif. With the decrease in PriceDif, Success_Rate increases notoriously.

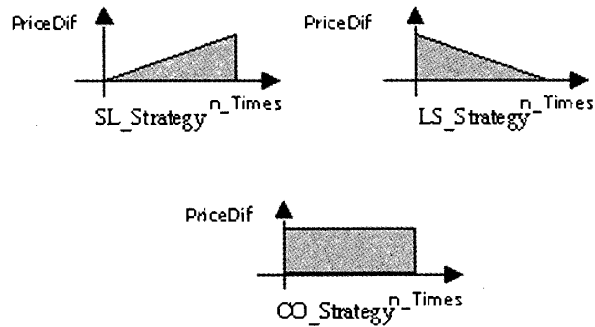
3.2 Seller-driven Change_PriceDif_Strategy Simulation

In this Seller-driven Change_PriceDif_Strategy, n_Times (i.e. the number of counter-proposals allowed before the breakdown of negotiations) and PriceDif for the Seller are changed. The Buyer's PriceDif is maintained constant and Strike_Price is randomly generated according to a normal distribution probability curve.

The following negotiation strategies (depicted in Fig. 3) are studied:

- (1) CO_Strategy: CO is the abbreviation for Constant, so PriceDif is maintained constant, as n_Times varies.
- (2) SL_Strategy: SL is acronym of “Small to Large”, so PriceDif increases from small to large, directly proportional with n_Times, following a linear curve.
- (3) LS_Strategy: LS is acronym of “Large to Small”, PriceDif decreases from

Fig. 3 Change_PriceDif_Strategy

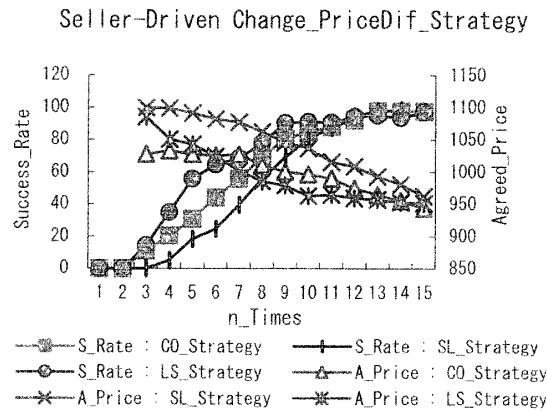


large. to small, inversely proportional with n_Times, following a linear curve
 n_Times is the number of messages exchanged before the negotiation ends. The results of simulation are shown in Table 2 and Figure 4. Note that in Table 2,

Table 2 Seller-Driven Change_PriceDif_Strategy

n	CO_Strategy		SL_Strategy		LS_Strategy		
	S_Rate	A_Price	S_Rate	A_Price	S_Rate	A_Price	
1	0.0		0.0		0.0		Zone_A
2	0.0		0.0		3.4	960.4	
3	9.0	954.4	0.0		24.8	990.0	
4	19.8	968.7	10.3	922.2	46.8	1003.9	
5	38.2	978.2	28.2	930.9	54.2	1020.2	Zone_B
6	50.2	993.8	37.8	947.7	72.4	1037.1	
7	60.8	1001.1	49.0	956.3	77.8	1054.6	
8	70.6	1015.0	64.3	990.2	79.0	1082.9	
9	72.4	1025.4	72.5	1012.7	85.4	1086.3	Zone_C
10	82.2	1035.2	81.7	1033.0	85.4	1075.2	
11	87.8	1036.7	92.2	1032.1	89.2	1096.1	
12	96.4	1053.4	94.0	1033.8	91.4	1071.0	
13	93.8	1058.2	98.8	1052.3	94.8	1079.7	
14	94.8	1042.9	98.4	1029.8	93.2	1075.5	
15	97.8	1057.0	99.6	1043.3	96.8	1087.3	

Fig. 4 Seller-Driven Change_PriceDif_Strategy



Success_Rate, Agreed_Price and n_Times are abbreviated S_Rate, A_Price and n, respectively.

From the simulation results, some observations are obtained. The change of Agreed_Price is insignificant compared with the change of Success_Rate. As the number of counter-proposal messages n_Times increase, we can see an inversely proportional relation between the curves representing the Success_Rate and Agreed_Price. By examining both curves, we can see the change of Agreed_Price is insignificant compared with the change of Success_Rate. In order to provide a more detailed analysis, we divide them into 3 zones, according to the value of n_Times:

Zone_A (n_Times : 1-5)

In this zone, the biggest variation (55.8%) of Success_Rate corresponds to LS_Strategy. This variation is near double of the maximum variation of CO_Strategy (30.8%) and SL_Strategy (18.2%).

Zone_B (n_Times : 6-10)

For LS_Strategy, the Success_Rate is more than 65% throughout the whole zone B and it is stable when compared with Zone A. With respect to SL_Strategy, the variation is 62.6%, which represent the biggest increase.

Zone_C (n_Times : 11-15)

In this Zone C, Success_Rate exceeds 80% for all of the 3 strategies. The biggest rise corresponds to SL_Strategy (19.2%) and is almost double of the maximum rise shown by LS_Strategy (6.6%) and CO_Strategy (12.8%). It is important to remark the fact that for the number of messages exchanged n_Times = 15, Success_Rate (99.4%) approaches 100%.

3.3 Buyer-driven Change_PriceDif_Strategy Simulation

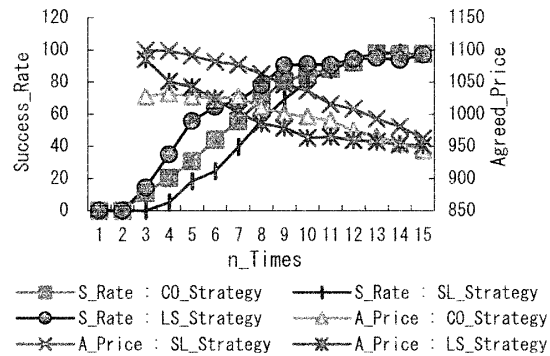
In this Buyer-driven Change_PriceDif_Strategy, the number of counter-proposals allowed denoted as n_Times and PriceDif for the Buyer being changed. The Seller's PriceDif is kept constant and the Strike_Price is randomly generated according to a normal distribution probability curve.

The same 3 negotiation strategies, i.e. CO_Strategy , SL_Strategy and LS_Strategy - explained before and depicted in Fig. 3 - are studied. n_Times is the number of messages exchanged before the negotiation ends. The results of simulation are shown in Table 3 and Figure 5. Note that in Table 3, Success_Rate, Agreed_Price and n_Times are abbreviated S_Rate, A_Price and n, respectively.

Table 3: Buyer-Driven Change_PriceDif_Strategy

n	CO_Strategy		SL_Strategy		LS_Strategy		
	S_Rate	A_Price	S_Rate	A_Price	S_Rate	A_Price	
1	0.0		0.0		0.0		Zone_A
2	0.0		0.0		0.0		
3	11.0	1027.2	0.0	1099.2	14.4	1085.3	
4	20.4	1032.5	5.2	1098.3	34.8	1050.8	
5	30.8	1027.4	18.2	1091.0	55.8	1042.2	Zone_B
6	44.2	1024.9	24.4	1081.7	65.4	1024.0	
7	55.8	1026.7	39.6	1077.4	68.4	1002.1	
8	69.4	1010.6	56.0	1061.9	78.0	985.3	
9	80.8	1000.2	69.2	1046.0	90.4	978.8	Zone_C
10	84.8	996.1	80.8	1036.5	91.2	962.9	
11	88.0	989.6	87.6	1015.3	90.6	964.3	
12	92.2	974.0	94.6	1008.2	94.4	959.5	
13	97.8	964.4	97.8	992.5	95.0	956.8	Zone_C
14	97.6	955.3	98.6	980.4	93.6	953.1	
15	97.6	943.7	99.4	962.2	97.2	951.2	

Fig 5 Buyer-Driven Change_PriceDif_Strategy
Seller-Driven Change_PriceDif_Strategy



From the simulation results, we can make the following observations. The Success_Rate and Agreed_Price are in a directly proportional relation. However, similar to the above studied cases, the change of Agreed_Price is insignificant compared with the change of Success_Rate. Moreover, regarding the change with n_Times, we can see the same tendencies of Seller-driven simulation, explained in subsection 3.2 above.

4. Concluding Remarks

From the outputs of the simulations performed on the selected negotiation strategies, we have noted several interesting properties.

First of all, regarding One_Shot_Strategy, common to both Buyer-driven as well as to Seller-driven, is the fact that the variation of Success_Rate is large, in accordance to the size of PriceDif. This means that the success or failure of dealings strongly depends on the PriceDif.

On the other hand, the change of Agreed_Price becomes insignificant compared with the change of Success_Rate. From Fig.2, we can note that in order to obtain 10% of profit increase, it might be necessary to give up a 80% decrease in Success_Rate.

From both Buyer-driven and Seller-driven Change_PriceDif_Strategy, regarding LS_Strategy - though the profit might be low - shows a higher success rate. In other words, LS_Strategy is preferable to use when one would like to have the negotiations steadily approved, although the profit is low.

Regarding SL_Strategy, the success rate is low at the early stages of the

negotiations, but if it succeeds it can raise a high profit. In other words, one might prefer to use this SL_Strategy in case of aiming higher profits even with some risks.

The last one to have a look at is the CO_Strategy. For this strategy, the variation in Success_Rate is constant, so the settings of factors other than PriceDif are directly reflected. This is, CO_Strategy is appropriate as a standard reference to set the other negotiations.

From the results of the simulations in ATRAS, we can understand precisely the behavior and results of the different negotiation strategies and the influence of the different parameters on the final outcomes.

Our future works include continuing simulations for cases which might appear in other electronic commerce scenarios.

References

- [1] Fujii, A., Okada, R., Oshida, N.; and Ohtani, T. 1999. Fresh Fish Trading System Development - A Kansei Engineering Approach. Kansei Engineering International journal. 1(1):75-80.
- [2] Chavez, A. and Maes, P. 1996. Kasbah: An Agent Marketplace for Buying and Selling Goods. In Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 75-90, London, U.K.
- [3] Tsvetovaty, M., Gini, M., Mobasher, B. and Wieckowski, Z. 1997. MAGMA: an agent-based virtual market for electronic commerce. Applied Artificial Intelligence.
- [4] Eriksson, J., Finne, N. and Janson, S. 1998. SICS MarketSpace - An Agent-Based Market Infrastructure. Workshop on Agent-Mediated Electronic Trading.
- [5] Kephart, J., Hanson, J., and Jairamesh, J., "Price-War Dynamics in a Free-Market Economy of Software Agents," Proceedings of Artificial Life VI, UCLA, The MIT Press, 1998
- [6] Okada, R., Fujii, A., Oshida, N.; and Ohtani, T. 2001. GEM: Goods-agents in Electronic Markets. In Proceeding of International Conference on Electronic Commerce 2001 (ICEC2001), Vienna, Austria.
- [7] Okada, R., Fujii, A., Oshida, N.; and Ohtani, T. 2002. Bounded-rational Agents

Representing Goods in Electronic Markets, In Proceeding of International Conference on Electronic Commerce 2002 (ICEC2002), Hong Kong.

[8] Bigus, J. and Bigus, J: Constructing Intelligent Agents using Java 2nd edition, 2001