A Personal e-Market Agent (PeMA)

Upmanyu Misra[1], Ushhan D. Gundevia[1], A.A. Basu[2]
[1] B.Tech., Computer Science & Engineering,
Institute of Engineering and Technology, Kanpur, India
{upmanyu_misra, ushhan}@rediffmail.com
[2] Lecturer, Department of Industrial Management and Engineering,
Institute of Engineering and Technology, Kanpur, India

Abstract. In this paper, we define a Personal e-Market Agent (PeMA) that can act both as a Buyer or a Seller Agent. It stores user profiles in a defined format and uses a combination of different market strategies to select the best bid available in a dynamic market. We outline the attributes required for the creation of a user profile and a methodology for the agent to adapt according to an ongoing negotiation of a bid and to select the best strategy for transaction (with other agents in the market) out of various strategies available.

1 Introduction

In the information economy of today, the plenitude and low cost of up-to-date information has enabled consumers to be better informed about products and prices. Search costs have reduced dramatically and a buyer can compare prices across various sellers of the same good. For example it is easy to compare prices of a book at various sites like Barnes & Nobles (www.bn.com), Amazon (www.amazon.com) and Borders (www.borders.com). Typically, the buyer visits a few sites and chooses the one with the lowest price. But the number of sites a person can visit is limited by the time he/she can invest in the monotonous job of searching. A better way would be to use a comparative shopping agent or a Shopbot for this purpose. Shopbots of today merely collect pricing information from vendors and then the human master chooses the party with whom to transact. These represent the simplest of agents. Agents, in general are specialized pieces of software that are semi-autonomous, suitable for a wide array of information processing tasks from doing simple exhaustive searches for a product's prices to negotiating complex transactions on behalf of their human owners.

As they evolve, agents will begin to interact, not just with human and static web-sites but also with one another. Interaction among agents would be supported by a number of efforts that seek to standardize the agent communication languages and interaction protocols, the myriad attempts to establish standard ontology for numerous products and markets, and the development of various methods for e-payments. In future, agents will invariably taking decisions on behalf of their human owners rather than being limited to the search job [1]. The ability of agents to process large amount of information and their ability to respond quickly to a change in the pricing environment would prove to be immensely useful. As competitive pressure and the need to respond quickly increases all parties (both consumers and producers) would begin using agents for carrying out transactions on their behalves. As more and more

Shopbots - buyer agents, come into play, sellers will have to rely increasingly on Pricebots - seller agents that can adjust prices automatically on the seller's behalf in response to market conditions. In such an agent-based economy, it would be both interesting and informative to observe how a group of seller and buyer agents behave when they have to achieve a certain goal.

In this paper, we purpose a framework for inter-agent communication. All our agents are economically motivated, and on being given task of purchasing/selling some items, they scan possible choices, and employ various strategies to find out the most optimal peer to transact with. A transaction in our model can be very general and it is possible to capture the associations and constraints between real-life objects in our model. We have also used our framework for carrying out simulations and have analyzed the interaction between Pricebots and Shopbots.

1.1 Related Work

There is a lot of work going on in standardizing agent communication languages, protocols, ontology's, etc. There is also work going on in developing methods for electronic payments and micro-payments.

Study of agent behavior, their impact on markets, and having agents employ intelligent strategies is being studied under information economics project at IBM T J Watson Research Center ([2], [3], and [4]). The focus is on analyzing and finding out the ramifications of collective behavior of agents. The most recent research in the direction of Strategies for sellers in an Online Auction has been recently published by IBM T J Watson Center [12]. Another very relevant work has been done in Fuzzy e-Negotiation Agents (FeNA's) [14] where the constraints are modeled as Distributed Fuzzy Constraint Satisfaction Problem (DFCSP).

Other projects in the area include those, which seek improvements in the quality existing markets through automated trading by software agents. To reap the benefits of market as effective instruments of resource allocation, automated trading by software is being used in non-traditional domains, such as fine-grained markets for electric power [5] and communication bandwidth, selling excess CPU time or storage capacity [6] etc. Also, the "Win or Learn Fast" (WoLF) principle [7] has been introduced recently, for varying the learning rate. Thus, these projects focus on the application of software agents in which economic intelligence has been infused.

1.2 Our Work

Our focus is two folds: on communicating strategies to an agent, and on complex price functions. In our knowledge, we are the first to introduce the concept of complementary goods, substitute goods and the price quotes as a function of time. We are not aware of any other research where an agent can buy "similar" goods in place of the goods that it is employed to buy. Our agents can also offer different price depending whether the seller is able to sell only few goods to us or everything to us. For example, if we are trying to buy chairs and tables, we may offer a different price if the seller only sells one thing, vis-à-vis another seller who is willing to sell us both. Also, agents may increase or decrease prices with time, according to a function supplied by the human master.

By incorporating the ability to express these real world constraints and having a more general concept of transaction (many goods can be specified in a single transaction), the agents can arrive at an optimal transaction from a much larger solution space. Finally we have simulated the behavior of a particular class of agents as described in [2]. We haven't come across any existing methodology that takes all these options into account while negotiating.

2 Requirements Specification

Transaction takes place when a buyer and a seller agree to exchange some commodity or a collection of commodity at a mutually agreeable price. For any transactions through agents, the following are essential:

1) Users must be able to specify their constraints and their strategies to the agent.
2) The agents must be able to communicate with each other in a meaningful way.

This is possible only when all of the following three criteria are met.

a) The agent should be able to locate and exchange messages with other agents.
b) The content (knowledge) that they share with each other must be in a format that is understood by all.
c) The words they use to communicate with each other should be understood and interpreted unambiguously by each of them (i.e., they must have a common vocabulary or ontology).

Specifying constraints and strategies to agents is a very difficult problem, as it aims to capture economic behavior of human beings in terms of a few parameters. We have confined ourselves to only those requirements, which we think are essential in specifying the economic behavior of agents, and which do not have a high computational cost. The specification mechanism should take care of following things:

- It should be possible to specify some of the relationships that exist in the real world between different items (e.g., whether the goods are complementary or substitute goods). Speaking more generally, users should also be able to specify their constraints on dealings as a function (e.g., the buyer might specify a constraint that the number of chairs to be bought should be exactly six times the number of tables. This can be done by specifying a function).
- It should be possible for the user to specify the price as a function of number of items. Usually, a person expects a discount when buying a large number of items. On the other hand, sometimes a person is willing to pay more if it can get a larger number of items from one place.
- It should be possible for the users to specify the price quote (their valuation) as a function of time. For example, a buyer may indicate to the agent that the need is at a certain time, so the agent would try to negotiate heavier discounts in the beginning, but would be willing to pay a higher price as that deadline nears. Sellers may also have asked their agents to increase or decrease prices with time.

For 2(a), there must be a discovery protocol to allow the agents to discover each other. Also, agents must be able to exchange messages with each other. For 2(b), there must be a common language that agents must understand. For 2(c) there must be a common vocabulary of terms. There are some standards for each of the above like KQML (Knowledge Query Manipulation Language) protocol [8], KIF (Knowledge Interchange Format) [9] and Ontolingua (the common conceptualization of the world) [10].

3 Design

Transaction in our design is made possible by the interaction of three types of agents - Buyer agents, Seller agents and a Central agent. The central agent acts as a matchmaker or a broker that maintains a repository of information about seller agents and helps buyer agents discover compatible seller agents. The agents in our design can negotiate multilaterally in a peer-to-peer framework (a buyer agent can talk to several seller agents and vice-versa) for completing a transaction. The transactions are modeled in the same general framework as discussed earlier. Buyers can specify for each item: the minimum quantity (min), the maximum quantity (max), price as a function of quantity, and price as function of time. Additionally, the relationship between different goods, the synergies between different goods, the strategies to be followed while negotiating, and the deadline by which such a deal should be completed could be specified.

3.1 Profile Building

First, the user creates a configuration file that contains the information the agents needs to know for carrying out transactions. After a buyer/seller agent is invoked, it generates a unique name for itself so that its identity can be maintained across the many sets of messages, which it may exchange with other agents during a single session. The buyer/seller agent then parses the configuration file and initializes its data structures.

3.2 Biding and Alternate Biding

The data stored in the configuration file cannot be directly used to carry out a transaction. This is because the data is in the form of a set of inequalities. For example, take a simple configuration file containing specification about two items, tables and chairs. Let the minimum number and maximum number of tables specified be 2 and 4 respectively and these numbers in case of chairs be 13 and 19 respectively. Now, any of the 21 (=3*7) possibilities would satisfy the requirements of the agent (user). But, if more constraints, like the number of chairs should be strictly equal to six times the number of tables, are specified, there remains only one possible tuple - (3 tables and 18 chairs) that satisfies the requirements. There is also a possibility that no solution satisfies all the constraints placed by the user. Such information is not immediately obvious from the format in which the user makes specifications. Thus, it is more practical to convert the specifications given by the user into set of tuples that satisfy these constraints. It basically corresponds to finding integer solutions to a set of inequalities.

Let there are n items, $i_1, i_2, \ldots, i_n$, then there are 2n inequalities, corresponding to:

$$minimum_j <= i_j <= maximum_j, \ 1<=j<=n.$$

Also, there will be an equality corresponding to each relation mentioned in the global constraints. From these inequalities, finding all possible sets of tuples that satisfies all the constraints is termed as getting the alternate bid representation. Thus the alternate bid is a set - it may contain more than one bid, each bid representing terms of a possible transaction. We use bid and tuples interchangeably but there is a slight difference; a tuple only represents the quantity of each item to be traded, whereas a bid also contains the valuation that an agent offers or is ready to accept, corresponding to each tuple. The alternate bid is in fact a set of bids and contains information about all possible transactions along with the acceptable price, which the agent might indulge with the peer.

3.3 Central Server

Seller agent contacts a central server that keeps information about all seller agents. A seller agent informs to the central server about all the objects that it intends to sell, the IP address and the port number it is listening on, and the time until which it would be willing to carry out the transaction. The central agent will either create a new entry, or update an old one, as the case may be. The message from a buyer agent to the central server consists of a list of items that it is interested in buying. The central server than returns a list of seller agents.

3.4 Negotiation

Now, the buyer and the seller agent are ready to exchange messages and transact. The transaction consists of a set of proposals and counter proposals, also called bids. The negotiation consists of two phases. In the first phase, from the set of possible transactions that satisfy their constraints, the most optimal transaction is chosen by the buyer agent. (Choose the exact set of goods to be purchased etc.). In the second phase, the negotiations on the price take place. The two agents try to arrive at a mutually acceptable price for the optimal transaction discovered in the first phase. Initially, the seller agents wait for a message to arrive. The buyer agent initiates the communication. The buyer agent tries to contact all the seller agents - as per the list achieved from the central agent, asking for the price quote of the goods it is interested in buying and sending along its bid (which goods it want to buy and quantity of each good to be bought). The price is not included because the buyer agent wants to know the seller's price before giving it own valuation. (This is also true of the real world transactions.) The seller agents, if they can satisfy buyer's constraints, respond to this proposal, with their bids. Out of the bids received, the buyer agent chooses the best one, based on the price.

In the second phase, both the parties check if the valuation in the bid it received is acceptable to it or not. If it is not, it makes the bid with the next price. The next price an agent offers may be a function of its bidding strategy, the previous price it offered to the same agent, the price that the peer is now offering and the prices present in the bid it has received from its other peers. This process ends when either an OK message (indicating that the transaction is committed) or a Withdraw message is exchanged.

Any message, except the first one, intended to get the price quote from the seller agent and its reply represents a firm commitment until the validity period expires. So on receiving a bid from its peer, the agent has to check whether the proposal it sent is still valid and the tuple present in the bid it has received is indeed present in the alternate bid it had sent (i.e. whether it had actually made the offer or not). Also, to maintain the consistency of the messages, at any time instance, the agents send only those proposals that they are able to fulfill. The agents at all time maintain information about the available resources that they have with them. This amount has to be reduced if they send a bid, and has to be increased whenever a negotiation fails (on receipt of Withdraw message).

3.5 Configuration File

The configuration file contains the following information:

- **Items**: objects to sell/buy and their properties. (Minimum and Maximum quantity, price as a function of quantity and price as a function of time). There may be multiple such entries - as many as the number of distinct items to be traded.
- **Global Constraints:**
  - **Capture synergy between objects:** expressed as a percentage on how much more the user is willing to pay above her normal valuation of these items. For example, if a user values a chair at 5 units and a table at 15 units and the synergy is 50%, then she is willing to pay 30 units for a combination of a chair and a table.
  - **Complementary goods:** for e.g., the number of tables must be exactly six times the number of chairs.
  - **Substitute Goods**: for e.g., the total number of tea and coffee packets to be exchanged must be ten.
- **Strategy:** the negotiation strategy to follow. Several strategies may be implemented; each identified by a number.

3.6 Message Format for Exchanging Proposals

It consist of four parts,

- **Name:** unique identifier to identify an agent.
- **Message type:** Buy/Sell/OK/Withdraw.
- **Alternate bid:** In the first phase it consists of set of bids that satisfy all the constraints. In the second phase, it consists of just a single tupel.
- **Date:** time until which the bid is valid.

4 Simulations

Our platform offers us a powerful framework to model different types of transactions and to carry out simulations. As a beginning, we have considered a simplified dynamic pricing model and have tried to implement the behavior of agent's [2]. We consider a simple market in which S seller agents compete to provide B buyer agents with a commodity, such as a specific book.

4.1 Strategies for Buyers

A buyer purchases a good if her valuation is more than the seller's price. In the process, if it has to choose amongst the many seller agents that satisfy its criterion (i.e., the price offered is less than buyer's valuation), it follows one of the following strategies:

1) **Random Buyer**: buyers receive offers from several sellers, picks out one that offers prices that are less than its valuation and then selects a seller in random from this lot. (Normally, one would buy from the first seller whose price is acceptable to us, without comparing it with other offers).
2) **Bargain Hunter**: buyer checks the offered price of all sellers, determines the seller with the lowest price, and if this price is lower than its own valuation, then purchases the good.

The buyer population consists of a mixture of buyers employing one of these strategies, with a fraction w_A using the Random Buyer strategy and a fraction w_B using the Bargain Hunter strategy, $w_A + w_B = 1$.

4.2 Strategies for Sellers

The seller agent follows one of the three strategies that we describe below. Each of the strategies requires agents to have different levels of knowledge about their environment (the buyers and other sellers).

1) **Game-Theoretic**: The GT strategy is designed to reproduce the game-theoretic equilibrium [11], provided that all sellers adopt it. It assumes full information about the buyer population, as well as about the competitor's prices or pricing strategies. In essence a game-theoretic agent assumes that all other agents abide by the prescribed game-theoretic strategy, and based on this assumption, it computes its own equilibrium price.
2) **Myopically-Optical**: The MY strategy makes use of all the information about all the buyer characteristics that factor into the buyer demand function, as well as the competitor's prices, but makes no attempt to account for competitor's pricing strategies. Instead, it is based on the assumption of static expectations: even if a seller is contemplating a price change, this seller does not assume that this will elicit a response from its competitors; instead it assumes that the competitor's prices will remain fixed.
3) **Derivative-Following**: The DF strategy is far less information intensive than other pricing strategies. In particular this strategy can be used in the absence of any knowledge or assumptions about one's competitors or the buyer demand function. A derivative follower simply experiments with incremental increase (or decrease) in its price, continuing to move its price in the same direction until the observed profitability level falls, at which point the direction of movement is reversed.

4.3 Implementation of Strategies

The buyer strategies are simple to implement. In case of a Bargain Hunter, a linear search of all responses was enough. In case of a Random Buyer, only a random number is to be generated. The seller agent strategies were implemented as follows:

1) **GT**: The game-theoretic equilibrium requires that one seller set its price to p* (equilibrium price), while the remaining seller all charge the monopolistic price - v (buyer's valuation). (p* is the equilibrium price calculated by game theory such that it is not possible for any sellers, charging v, to increase their profits by beginning to charge (p* - e), where e is a very small change in price). This computation does not address the question as to which seller volunteers to charge the low price p*. In our implementation, the first seller which notices that no other seller is charging p* sets it price to p*.
2) **MY**: The algorithm for calculating the next price offered by a myopically-optical seller is based on the following observation: when considering only static expectations (i.e., other seller do not change their prices in response to the price change by the seller), a seller can boost its profit either by moving to v or by undercutting the lowest price offered among all competitors (by obviously the smallest amount).
3) **DF**: The price increment is taken to be the minimum change possible in our framework. If a seller founds that profit in the previous interval was higher, the direction of change was left unchanged otherwise it was reversed.

4.4 Other Details

We performed simulations with v = 90, c = 70 and gt + my + df = 5; gt, my, df Є {0, 1, 4, 5}

where v is the buyer's valuation and c is the cost of an item for the seller. 'gt' denotes the number of the GT sellers, 'my' denotes the number of MY sellers, and 'df' denotes the number of DF sellers.

The maximum average profit that could be obtained per buyer by each seller would be 4. (The maximum profit is 20; divide that by the number of sellers.) The ratio of number of random buyers to the number of bargain hunters was kept at 2/3. The minimum price being 1, p* as referred to be in above, works out to be 70 + (20/69)*8, which was set at 72.

In Table 1, we provide a comparative summary of the results. Each entry represents the average profit of various sellers for each buyer. The first number is for the profits of four sellers following the strategy of that row. The second number is for the profit of a single seller following the strategy of that column. For e.g., let us have 4GT sellers and a single MY seller. From the table we can see that the 4GT sellers make an average profit of 1.6, while the MY seller makes a profit of 1.3.

Table 1

	1GT	1MY	1DF
4GT	(1.5, 1.5)	(1.6,1.3)	(1.4, 2.1)
4MY	(1.7, 1.3)	(2.3, 2.3)	(2.3,1.2)
4DF	(1.6, 1.4)	(1.4, 7.4)	(3.3, 3.3)

4.5 Analysis

In our simulations, 40% of buyers are random buyers, and they are equally distributed amongst all the five sellers. The remaining 60% go to the seller who is under-cutting everyone else. If there is more than one seller selling at the same low price, then these buyers get equally distributed amongst them. The pure combinations (5 GT, 5MY, and 5 DF sellers) lie along the diagonal of Table 1.

First, consider the 5 GT sellers. Here, an under-cutter can grab a significant market share (68%), but the price is such that it is not profitable to undercut. Below the price of ($p^* = 72$), undercutting is not worthwhile because the profit margins are so low that it is better to charge the monopolistic price $p_m = 90$ and accept the low 8% market share. Each GT Pricebot earns relatively low profits of 1.5, on an average, as compared to the theoretical maximum of 4 that could be obtained by a collusive cartel in which each seller charges the monopolistic price of 90.

Now consider the 5 MY sellers. As is clear from the figure, they undercut one another until the price falls to p^*, at which point undercutting becomes less profitable than charging the monopolistic price of 90. Thus price suddenly jump up to this level. However, as soon as this occurs, undercutting ones again become attractive, and the price war cycle begins anew. Although the MY sellers fall into endless price wars, there average prices are higher than those of GT sellers. This is reflected in their higher average profit: 2.3 versus 1.5 in this example.

Now consider the 5 DF sellers. Interestingly, although they are the least informed, they maintain the highest prices and therefore, the highest profits. Their average profit of 3.3 is very close to the optimal value of 4 that could be obtained by a cartel.

Next, by pitting different Pricebots against each other, we find to try the best of the three strategies

From the table, it is clear that a GT agent is unaffected by the strategies of the other agents. They either operate at v or p^*, while other strategies mostly operate in the middle. The GT-agent operating at p^* will garner all the markets of bargain-hunters, while others will get equal share of random buyers. The GT-agent operating at v will get only its share of random buyers. There can only be transient effects if agent employing others strategies also start operating at v, though they will move away from it immediately.

When an agent is pitted against 4 DF's, it fares many betters than does a DF. MY is more effective because it undercuts the 4 DF's by the minimal amount necessary to grab the 68% market share. For the same reason, even 4 MY agents on an average perform better than a DF agent.

Thus, if agents were permitted to select pricing strategies on the basis of expected long-tern payoff; a society of 5 DF's will be unstable. The first agent to reconsider its strategy choice would switch to MY, as would each successive agent until all were converted to MY. The situation is analogous to Prisoner's Dilemma [11]: self-interest compels all of the sellers to defect to the MY strategy, even though this leads to a lower profit than would be obtained if they were to all adhere to the DF strategy.

Of all the three strategies considered, MY should be the algorithm of choice whenever detailed buyer information is available. Otherwise, a simpler strategy like DF may be the only choice. In practice the detailed buyer information required by MY and GT is unlikely to be obtained very easily. A seller following MY strategy would be better off if it re-prices faster.

The seller agent should have more foresight. The algorithm will improve if the seller agent while deciding its next price could also take into account the anticipated pricing behavior of its competitors. One promising method is Q-learning [2].

5 Conclusion

Building economic behavior in agent is necessary so that the agent can transact with each other and with other human beings in the same way as we humans transact with each other. But, before indulging in an all-agent economy, we need to stimulate agent's behavior and find out the ramifications when all of them follow pre-defined strategies. Or else, wide chaotic economic swings like cyclical price wars could become more frequent (since agents are not subjected to the restraints that normally rein economic activity - their transactions takes place almost instantaneously, cost is negligible, and distance is negligible, and distance is irrelevant).

Ours is an attempt to the above. We have tried to model complex human transactions - we allow the user to specify a set of items for a transaction instead of specifying each item in a separate transaction and as a result, the optimal transaction can be chosen from a larger search space. We have stimulated the interactions of Shopbots and Pricebots in a dynamic posted pricing environment. The results show that collective behavior of agents may not resemble that of human. Learning does not come intuitively to them, and as a result they may be involved in cyclical price wars.

That to evolve, agents will have to learn, adapt, and anticipate is a statement that needs but little reflection to agree with. For this, they would have to use a variety of machine learning and optimization techniques. Further work needs to be done in this direction and the complexity here is in learning, when the environment is constantly changing. The challenge is to have buyer and seller agents mimic the complex behavior and strategic thinking of humans and then, explore and analyze the resulting interactions among them.

Finally, agent interaction in our framework can be viewed as that taking place among faceless agents. The agent does not have long-term identity. As a result, there are no inter-temporal transactions. Faceless agents cannot take upon themselves obligations for future delivery of goods or money. Therefore, the problem boils down to, "how does an agent learn the abstract representation from experience, how does it learn environment models in these representations, and furthermore how does it learn models for many different ways of behaving from a relatively small amount of experience (called off-policy learning)". [13]

References

1. R. Guttman and P. Maes, "Agent-mediated integrative negotiation for retail electronic commerce." in the Proc. Of Workshop on Agent Mediated Electronic Trading (AMET' 98), 1998.
2. J. Kephart, J. E. Hanson, and A. R. Greenwald, "Dynamic pricing by software agents," Computer Networks, 2000, available from URL: http://www.research.ibm.com/infoecon/paps/rudin.ps.gz
3. A. R. Greenwald and J. O. Kephart, "Shopbots and Pricebots," in Proc. of the 1st ACM Conf. On Electronic Commerce. Nov. 1999, ACM Press.

4. A. R. Greenwald, J. O. Kephart, and G. J. Tesauru, “Strategic Pricebot dynamics,” in Proc. of the 1st ACM Conf. On Electronic Commerce. Nov. 1999, ACM Press.
5. H. Green, “Good-bye to fixed pricing?” Business Week, May 4 1998, pages 71-84.
6. Ori Regev and Noam Nisa, “The popcorn market – an online market for computational resources,” in Proc. of the 1st Intl. Conf. on Information and Computation Economics (ICE’98), Oct 1998.
7. Michael Bowling and Manuela Veloso, “Multiagent Learning using a variable learning rate”, Artificial Intelligence 136(2002) 215-250, Elsevier.
8. “KQML: An agent communication language,” http://www.csee.umbc.edu/kqml
9. “KIF: Knowledge interchange format,” http://www.csee.umbc.edu/kif.
10. “Ontology,” http://www.csee.umbc.edu/kse/ontology
11. J. Nash, “Non-cooperative games,” Annals of Mathematics, vol. 54, pp. 286-295, 1951.
12. Xin Guo, “An optimal Strategy for Sellers in an Online Auction”, ACM Transactions on Internet Technology, Vol. 2, No. 1, February 2002, Pages 1-13.
13. Satinder Singh “Reinforcement Learning and Hierarchical Learning”, 3rd Special Issue on Machine Learning, Kluwer Academic Publishers, 49, 107-109, 2002.
14. Ryszard Kowalczyk, “Fuzzy e-Negotiation agents”, Soft Computing 6 (2002), Springer-Verlag, pg. 337-347.