The Design of Utility Functions for Information E-marketplaces with Price Quote Fees

Content Areas: multiagent systems, web agents, market-oriented programming

Abstract

Information e-marketplaces enable entities to buy and sell information; these buying and selling entities can be humans, or automated agents that represent them. In this paper, we analyze the characteristics of information e-markets and the behavioral character of the autonomous agents that operate in these markets. We describe four desirable characteristics of information sellers' utility, and present a specific definition of utility that exhibits these characteristics.

We continue by addressing the bandwidth problem that sellers face due to buyers and sellers that perform price sniffing. We show that this argues for a system that includes middle-agents along with buyers and sellers. Furthermore, sellers and middle-agents can protect themselves from the costs associated with multitudes of requests for price quotes (RPQs) by establishing a fee for those requests. This forces buyers and sellers to consider more carefully what prices are truly important to know, which in turn leads to the introduction of intelligent agents that will decide when it is profitable to look for additional price quotes.

1 Introduction

This paper focuses on a specific type of electronic marketplace, the information e-marketplace, where the commodities being bought and sold consist primarily of information such as that found in books, CDs, journals and magazines. Such information markets pose challenging questions regarding how to deal with information, how to sell it, and how to price it. Information is different from regular commodities since it does not need to have a single physical embodiment (i.e., it can be duplicated at virtually no cost), and the same information can be presented in various digital formats (i.e., the same commodity can have multiple instantiations).[1] Information can also be decomposed into smaller components, or amalgamated into larger collections, in which people may be interested. Operators might be applied to information modules, creating new products based upon buyers' requests or as new product suggestions to buyers.

In information e-markets with price quote fees, sellers may be overwhelmed by the number of "request for price quotes" (RPQs) coming from buyers who want to compare prices, and from sellers who want to be updated on the competition's prices. In this paper we address this difficulty, and examine its influence both on the agents themselves and on the behavior of prices. The sellers defend themselves from the burden of handling multitudes of price quotes by charging agents that want to get a quote. In our model, buyers that perform the transaction as a result of the price quote get the price quote fee returned. In such an e-market, buyers need to consider the potential value of the sellers' price quote, and how much it will increase their utility. Sellers that use other sellers' prices in order to set their own prices will have to consider the desirable frequency of performing those updates. Furthermore, sellers need to consider if they need essentially perfect information about other sellers' prices, or whether partial information is sufficient.

In addition to the technique of associating a fee, sellers can handle the multitude of RPQs by delegating this role to another agent: the middle-agent. The middle-agent, like the InfoCenter agent that Yarom et al. presented in [Yarom *et al.*, 2002], can handle the transaction with buyers. In this model, sellers will have to sell to the middle-agent, which will reduce the number of requests to sellers. The InfoCenter will have to be prepared to handle buyers' requests. Both sellers and InfoCenters may charge a fee for the RPQs in order to reduce the number of price requests.

Section 2 presents a detailed investigation of the dynamics of these markets including InfoCenter agents. We lay the groundwork for simulations in Section 3, followed by the experiments and their results presented in Sections 4 and 5. We conclude in Section 6.

2 The Model

We used [Yarom *et al.*, 2002] model for information marketplaces that contain sellers, buyers, and Information Center

[1] For example, printed information can be represented in PDF, Postscript, eBook or ePaper formats, audio information in WAV or MP3 formats, and video information in DVD, streaming video or AVI formats. Since electronic information is easily duplicated, special copyright protection may need to be considered (e.g., see work by [Ketchpel *et al.*, 1997]); in this paper, we assume that the copyright issue is handled by the marketplace.

agents (InfoCenters). InfoCenters are software agents that act as information intermediaries, and can reside, for example, in a library, at a portal Web site, or at a site that answers user questions.

Yarom et al. [Yarom *et al.*, 2002] shown that by extending the marketplace model proposed in [Kephart *et al.*, 2000] with automated agents that serve as middlemen and Information Service Providers, we achieve a marketplace where all participants obtain an increase in their utility. The profit of each participant in the market was positive and higher than in marketplaces with no Information Centers. Moreover, our previous results indicated that the addition of Information Center agents to an information e-marketplace leads to a decrease in price-wars, and therefore to a more stable market.

In general, information commodities are offered by S sellers, and may be bought by any of the B buyers (we assume $B >> S$). Each buyer generates purchase orders at random times, at a rate of ρ_b, while each seller resets his price at random times, at a rate of ρ_s. The worth of a good to a buyer b is represented by the value V_b. The cost of production for a seller s is C_s.

In this paper, we consider two specific aspects of information marketplaces and their effects on the design of trading agents. Our results further emphasize the desirability of introducing middlemen agents, such as InfoCenters, into the marketplace.

First, we focus on the automated trading agents' utility function design. We analyze the impact that different utility functions implemented by automated agents could have. A discussion of these choices is followed by a desiderata of four properties that automated sellers', buyers', and InfoCenters' utility should have. Those properties are desirable when determining a utility function to properly describe the performance of sellers, buyers, and InfoCenters. We then analyze aspect of the automated sellers' and buyers' behavior when these agents are required to handle requests for price quotes (RPQs) at different loads. Our results show that strategically designing automated sellers and buyers leads to better handling of the RPQs when an additional fee is charged for obtaining these quotes.

2.1 Desiderata for Automated Trading Agents' Utility Function

Seller's Utility

We are interested in understanding the effect that different utility functions may have on sellers' strategic behavior in an information marketplace. Since Information Center agents play the role of sellers when they offer new information products to buyers, this analysis will also shed light on how InfoCenter agents should be designed.

In a simple e-market that contains S sellers and B buyers, assuming $B >> S$, a simple and reasonable utility function for seller s is the following:

(1) $U_s(t) = \Sigma profit_s(t)$ when $s \in S$.

The problem with utility function (1) is that as long as time t increases, utility increases as well. We would prefer that for a large enough time period T, we would have, for each $t1, t2 \in T$, that $U_s(t1) = U_s(t2)$. For simplicity of analysis it is desirable to manipulate time-independent utility functions. That is, we would prefer to be able to arrive at conclusions regarding the sellers' utilities that are not dependent on time.[2]

Therefore, we might consider an alternative utility definition (2):

(2) $U_s(t) = \frac{\Sigma profit_s(t)}{trans_s(t)}$ when $s \in S$ and $trans_s(t)$ is the number of transactions performed by seller s.

This function's weakness is that it is not monotonic in total profit. A utility function is considered monotonic in total profit if the utilities of different sellers can be compared, and this comparison teaches us which seller's profit is higher. Utility function (2) considers the average profit and not the total profit, and therefore it cannot be monotonic in the total profit.

The following example provides a clarifying explanation. Assume a marketplace that consists of two sellers, one hundred buyers, and a single information commodity I_1. The cost of this commodity is zero for both sellers. The buyers are distinguished as follows: 50% of the buyers compare prices prior to their purchases, and the other 50% of the buyers choose a seller in a random way. The information commodity has a value of one for each buyer. Assume that seller s_1 sells I_1 at a fixed price of 0.5 and seller s_2 sells it at a fixed price of 0.75. Evaluating the market after 1000 transactions shows that the 50 buyers who are price competitive buy I_1 from the cheaper seller (s_1) and the other 50 buyers buy half of the time from seller s_1 and the other half of the time from seller s_2. All the sellers perform a transaction at the same rate (ρ_s). Therefore, after 1000 transaction, 500 transactions will be performed by the 50 price comparative buyers and the other 500 transactions will be performed by the other 50 buyers. Seller s_1 will perform the 500 transactions with the price comparative buyers. The 500 transactions of the buyers that do not compare prices will be split between the two sellers, 250 to each. As a result, seller s_1 will perform 750 transactions, while seller s_2 will perform 250 transactions.

The utility of seller s_1 according to (1) is $U^{(1)}_{s_1} = \Sigma profit_{s_1} = 750 * 0.50 = 375$ and according to (2) the utility is $U^{(2)}_{s_1} = \frac{\Sigma profit_{s_1}}{trans_{s_1}} = \frac{750*0.5}{750} = 0.5$. Similarly, the utility of seller s_2 according to (1) is 187.5 and according to (2) the utility is 0.75. Hence, seller s_1 attains a higher profit than s_2, but it also attains a lower average profit than seller s_2's average profit. Another option to consider is the utility function (3):

(3) $U_s(t) = \frac{\Sigma(profit_s(t))}{\frac{1}{S}trans_S(t)}$ when $s \in S$ and $trans_S(t)$ is the total number of transactions performed by all the S sellers[3].

Using the example above, the utility (3) of seller s_1 is $U_{s_1} = \frac{750*0.50}{\frac{1000}{2}} = 0.75$, and the utility of seller s_2 is 0.375.

[2] This seems like a particularly reasonable assumption with digital products, whose sale does not free up shelf space.

[3] We assume that this information is available in the market.

Furthermore, the utility remains the same for any number of transactions.

From the above analysis, we suggest that there are four properties we would like automated sellers' utility functions to have:

1. Time independence — for a large enough period of time T, for each $t1, t2 > T$, $U(t1) = U(t2)$.
2. Monotonicity in the profit — the profits of sellers can be compared, that is, if the profit of a seller s_1 is higher than the profit of a seller s_2, then $U_{s_1} > U_{s_2}$.
3. Monotonicity in the transaction — if seller s_1 had performed more transactions when compared to another seller s_2, and for each transaction of seller s_2 there is a transaction of seller s_1 that yielded the same profit (i.e., $profit = price - cost$), then $U_{s_1} > U_{s_2}$.
4. Normalization — if the prices of a transaction are given by a value $p \in [0, 1]$, then for every constant $a > 0$ the utility values will be $U \in [0, a]$.

It can be shown that the utility function presented above in (3) follows the four desiderata for sellers as well as for InfoCenter agents in an information marketplace as described here. In this paper, when we refer to the utility function of the sellers or the InfoCenter agents we mean the utility function specified in (3).

Buyer's Utility

The utility of buyer b at time t, after it has bought r products at a price P and cost C is $U_b(t) = v - \frac{\Sigma_{i=1}^{r}(P(t)-C(t))}{\frac{1}{B} trans_B(t)}$, when $trans_B(t)$ is the total number of transactions performed by the B buyers at time t. v denotes the value of one commodity for the buyer. We assume that all basic commodities have the same value of 1. A newly-produced information product will have a value that depends on the information content of this new product. For example, if a new information product was created by combining two basic information pieces, then the new product will have a value of two.

The same properties desirable for the sellers' utility function are also desirable for the buyers' utility functions: time independence, monotonicity in the profit, monotonicity in the transactions, and normalization.

2.2 Handling Price Quotes

In an information marketplace, buyers and sellers can generally compare prices automatically. This comparison aids buyers in finding cheaper sellers, and it aids sellers in setting more competitive prices. Nevertheless, asking for price quotes and responding to requests for price quotes impose an additional load on agents acting on behalf of sellers; this additional load needs to be considered at the design stage of these agents.

In practice, this additional load causes sellers to increase their bandwidth. Notice that an increase in requests for price quotes does not necessarily increase the number of transactions, especially in cases where the requests come from other sellers that are learning about prices in the market. In these cases, sellers will just suffer from higher costs. We now consider the impact of taking into account this load while designing strategic behavior for automated sellers in information marketplaces.

We first define the number of price quotes that sellers may have in an e-market composed only of sellers and buyers. The sellers can receive RPQs both from buyers and from other sellers. Therefore, the load of price quotes for a seller s is given by $load_s = extra\_load_s + trans_s$ when $extra\_load_s(t) = ExReq_B(t) + ExReq_S(t)$. $load_s$ denotes the total load that a seller s has, which is a combination of the RPQs initiated by buyers that perform transactions with seller s ($trans_s$), and other RPQs ($extra\_load_s$).

As we assumed in our model, there are B buyers which perform a transaction at a rate of ρ_b. Therefore, the probability for one of the buyers $b \in B$ to perform a transaction is $\rho_B = \Sigma_b \rho_b$. Each w_i in the vector $\vec{w}$ ($\vec{w} = (w_1, .., w_n)$) stands for the fraction of buyers that compare i prices. Thus, the probability that a buyer that performs a transaction will approach a seller is $\Sigma_{i=1}^{S} w_i \frac{i}{S}$.[4] The probability that buyers will approach a seller for a price quote and will not perform the transaction (will choose one of the other $i-1$ sellers) is $(1 - \frac{1}{i})$. Therefore, $ExReq_B(t) = \rho_B B \Sigma_{i=1}^{S} w_i \frac{i}{S}(1 - \frac{1}{i})$.

We stated that each seller performs a price update at the rate of ρ_s, and $\rho_S^{s_1} = \Sigma_{s \neq s_1} \rho_s$. We define $\overrightarrow{x^{s1}} = (x_0^{s_1}, .., x_k^{s_1})$, when each $x_i^{s_1}$ denotes the fraction of sellers excluding s_1 that compare i prices when setting their price. The number of price quote requests that a seller s_1 obtains from the other sellers at time t is: $ExReq_{s_1}(t) = \rho_S^{s_1}(S-1)\Sigma_{i=1}^{S-1} x_i^{s_1} \frac{i}{S-1}$.

Two ways in which sellers can handle the load of RPQs received are: (1) by charging a fee for providing a price quote, and (2) by approaching an intermediate agent that will handle the buyers and their requests for price quotes.

If the seller chooses to charge a fee for getting a price quote, then this will lead to a new information economy. In this economy, sellers and buyers consider the *benefit of knowing* the sellers' prices. For example, assume that there is a buyer that wants to buy a book and it is willing to pay up to $50 for it. Assume that this book is being sold at prices in the range of $29.99 – $34.99, and that getting a price quote costs 10 cents. Then, comparing 10 sellers' prices will cost the buyer $1, when the potential saving is $5 ($34.99-$29.99). Therefore, comparing 10 sellers can reduce overall spending, but comparing the prices of 100 sellers will not (it costs $10 which is more than the potential saving). Moreover, since the buyer is willing to pay $50 for this book, then maybe paying $29.99 or $34.99 is not that significant. In this paper we study this problem by implementing an intelligent buyer in Section 2.3.

Another possibility by which sellers can handle the costs of the requests for price quotes is by interacting with intermediaries like the InfoCenter agents in addition to the buyers in the market. We introduced the notion of intermediaries in information e-markets in [Yarom *et al.*, 2002]. These InfoCenter agents can not only buy and sell information, but also

[4] The probability that seller s will be one of the i sellers that the buyer will approach out of all the possible i sellers is $\binom{S-1}{i-1} / \binom{S}{i} = \frac{i}{S}$.

can procure and sell manipulated (i.e., processed) information.

The benefit of a full-fledged economy is that sellers will need to interact mostly with InfoCenters, and in that way they will reduce the load of handling RPQs (that will be managed by the InfoCenters). Buyers will buy from InfoCenters, which will need to be prepared to handle them and their requests. Furthermore, sellers and InfoCenters can collect a fee for RPQs, in order to decrease the motivation to perform price sniffing.

The seller can predict its expected load by computing the function $load_s = extra\_load_s + trans_s$.[5] The seller can use this information with the cost of supporting such bandwidth (e.g., servers) and to set a fee to cover some or all the costs of supporting it. Furthermore, if the sellers work only with the InfoCenters, then their load will be reduced by $ExReq_B$. Since $B >> S$, this is the significant part of the load. In other words, moving from a buyer-seller economy to a full-fledged economy with buyers, sellers, and InfoCenters reduces the load of sellers' price quotes significantly. The InfoCenters, on the other hand, will have to recoup the cost of handling the buyers' requests by using a fee.

2.3 Intelligent Buyers and Sellers

Agents can benefit in information markets with price quote fees, if they consider which prices are important to know. Intelligent buyers can consider the potential increase to their profit of obtaining additional price quotes. Intelligent sellers and intelligent InfoCenters can consider the increase in their profit when setting their price by knowing the other sellers' and InfoCenters' prices. The rest of this section describes the behavior of the intelligent buyers and the intelligent sellers and InfoCenters.[6]

An intelligent buyer needs to choose between performing the transaction with the prices that are currently available to it, or asking for additional price quotes. We should remember that if a transaction is performed with a seller from which an RPQ was requested, then the fee is returned to the buyer. Therefore, in the beginning, before the buyer has any price quotes, it chooses a seller randomly and it asks for a price quote. If the buyer has at least one price quote, then it will look at the value of $v - (min_{price})$ (where min_{price}=minimum price available and v is the buyer's value of the information product). If this value is large enough then the buyer will perform the transaction. The buyer can rely on its experience to decide what value is big enough. Additionally, when the buyer has more than one price quote (i.e., $num(quotes) > 1$) it can try to guess the potential profit of requesting an additional price quote. The buyer compares the potential increase in its profit by asking for an additional price quote and the RPQ's fee using the following rule: $\frac{max_{price} - min_{price}}{num(quotes)} > fee$.

[5] In real e-markets, the seller can use the history logs for predicting the expected load.

[6] This reasoning about when to explore prices has many similarities with the notion of *metareasoning* and *exploration vs. exploitation* in the AI literature; see, for example, [Russell and Wefald, 1991; Carmel and Markovitch, 1999].

Intelligent sellers and intelligent InfoCenters face similar problems when deciding how to handle RPQs. We explain this for sellers; the same reasoning applies for InfoCenters. The problem that sellers need to handle is to decide when the other sellers' prices will increase their own profit when setting their price accordingly. First, the sellers can use pricing algorithms that do not consider the other sellers' prices, like GT [Kephart *et al.*, 2000] (explained below in Section 3.1) and DF. If sellers apply pricing algorithms that do consider other sellers' prices, like MY, then sellers need to consider the optimal frequency (ρ_s) of updating their prices.

Sellers can benefit even more by using their profit value in order to decide when to look for other sellers' prices. Sellers can use the DF pricing algorithm as long as the profit levels are high. When the profit levels become lower than expected, the sellers will use the MY pricing algorithm in order to set the price. Then they will return to using the DF pricing algorithm until the profit falls again. In that way, sellers perform price sniffing only when they believe that they can increase their profit significantly. In order to decide what is a desired price level, sellers can use the MY pricing algorithm for a period of time for monitoring profit levels. Sellers can use this profit level as a desired profit level value.

3 Simulation Settings

We have empirically tested the impact of RPQs on an information e-market. One simulation consists of a series of repeated encounters between finite sets of buyers, sellers, and InfoCenters. A finite set of basic commodities is offered for sale by the sellers. New commodities can be created by InfoSPs and can be sold by InfoCenters.

Sellers and InfoCenters offer information products that can be bought. Each product is initialized with a fixed price. Each seller holds an infinite amount of the products offered. The cost of producing a basic commodity is zero.[7] During one simulation, the price is updated according to the sellers' strategies at a given rate ρ_s. The buyers choose a seller, based on the products that they are interested in and based on their strategy (as explained below). Buyers approach sellers at a rate ρ_b. Once a buyer approaches a seller, the transaction is necessarily performed between the two.

3.1 Buyers' and Sellers' Strategies

Buyers need to choose from which seller they will buy the commodity of interest. We have examined four algorithms that were implemented by information consumers (the first three algorithms were studied in a simpler market by Kephart et al. [Kephart *et al.*, 2000] and in an information marketplace with InfoCenters in [Yarom *et al.*, 2002]). Here, we present new results from studying the effects of charging fees for obtaining price quotes. The numbers in parentheses represent the percentage of such buyers in our tested market:

1. Compare-All (60%).
2. Compare-None (10%).

[7] A commodity created after applying an operator by the InfoSP incurs an additional cost.

3. Compare-two (20%) — Each buyer chooses two information sources randomly and then buys from the cheaper one.
4. Intelligent Buyer (10%) — This is the behavior described in Section 2.3.

Information suppliers in the marketplace apply four algorithms for changing the price of their commodity (the first three were studied in a simpler market [Kephart *et al.*, 2000] and also in a market including InfoCenters [Yarom *et al.*, 2002]).

1. GT (Game Theory) — Each seller choose randomly a mixed strategy that is in Nash equilibrium using the following function $p(F)$, where F is a random value between the cost c of the commodity and its value v (in our case $F \in [0, 1]$). S denotes the number of sellers in the market, and w_i is the fraction of buyers that compare i prices. $p(F) = c + \frac{w_1 * (v-c)}{\Sigma_{i=1}^{S} i * w_i * (1-F)^{i-1}}$.
2. MY (Myoptimal) — The seller assumes that current known market conditions do not change, and it sets the price of the commodity it is willing to sell such that it maximizes its short-term profit. In order to be able to set a price myopically, a seller needs knowledge about the buyers' population, the number of competing sellers, and all of the sellers' prices.
3. DF (Deviate Follower) — The seller keeps increasing the price of a commodity as long as its profit increases. The seller decreases the price when its profit drops off a certain amount. The seller continues decreasing the price as long as its profit increases. When the profit starts to decrease and has passed a certain level, then the seller starts to increase the price.
4. DFwMY — The intelligent seller and InfoCenter may use either DF or MY as described in Section 2.3.

4 Experiments

The simulations we ran examined the effect of charging a fee for RPQs. In all of the markets studied, there were two basic commodities, three sellers, and one hundred buyers. We first studied an e-market composed only of buyers and sellers. We compared the effectiveness of the different pricing algorithms. The MY pricing algorithms compares prices in order to set the new price, while DF and GT do not. Therefore, two different MY pricing algorithms were tested, one that pays a fee for RPQs (MYF) and one that does not (MY). In addition, we tried to find the optimum of the rates of price updates (ρ_s) that would minimize the costs of RPQs.

In addition, we looked at the effect that collecting fees for RPQs has on buyers. The utility of buyers implementing each one of the buying algorithms described in Section 3.1 was compared.

Finally, as we showed in Section 2.2, InfoCenters reduce the RPQ load of sellers, and divide the load between sellers and InfoCenters. While InfoCenters have taken on the costs of handling the large number of RPQs received from buyers, we consider whether they still remain profitable when they pay a fee for providing RPQs, and when they do not.

	DF		GT		MY		MYF	
DF	0.46	0.46	0.14	0.34	0.25	1.03	0.25	1.02
GT	0.15	0.09	0.09	0.09	0.05	0.15	0.05	0.14
MY	0.46	0.09	0.15	0.07	0.47	0.47	0.47	0.46
MYF	0.45	0.09	0.14	0.07	0.46	0.47	0.46	0.46

Table 1: The sellers' profit when the fee is set to 0.1 (the left value is of the single seller, and the right value is of the two other sellers)

$1/\rho_s$	$U(S)$	$U(S)$ w/ fee	pricing	$U(IC)$
5	0.38	0.27	DF	0.34
10	0.41	0.29	GT	1.64
15	0.44	0.32	MY	1.48
20	0.44	0.33	MYF	1.47
25	0.45	0.33	DFwMY	0.28
DFwMY	0.50	0.48	DFwMYF	0.27

Table 2: All sellers apply MY

5 Results

First, we discuss the effects that paying fees for RPQs has on the sellers' profit. The DF and the GT pricing algorithms do not compare prices, while the MY pricing algorithm does compare the prices. As can be seen in Table 1, sellers gain the highest profit when using the MY pricing algorithm (1.03 when not paying fees, and 1.02 when paying fees). This behavior can be seen easily in the table, except in the case of 3 DF sellers that gain 0.46, while the 2 MYF earn 0.45 (in the configuration with 1 DF seller). But we should note that the homogeneous case, where all sellers use the DF pricing algorithm, is not stable, since the seller will prefer to use the MY pricing algorithm in order to increase the short-term profit (increase the profit from 0.46 to 1.03 in this case). Therefore, even though the two sellers in the 3 DF case earn more as compared to 2 MYF (with 1 DF case) (0.46 vs. 0.45), this configuration is not stable. Furthermore, sellers will benefit from implementing MY or MYF, since if one of them switches to GT or DF then its profit will decrease (the MY and MYF profits are 0.47 and 0.46 respectively, while moving to the GT and DF pricing algorithms will change the profit to 0.07 and 0.09, respectively).

We examine the most beneficial update ratio (ρ_s) for sellers that apply the MY and MYF pricing algorithms. We can see from Table 2 that higher ratios increase the profit of sellers both when they pay a fee and when they do not. This is because frequent price change enable the sellers to set their prices to the optimal price more often. Moreover, the intelligent seller (DFwMY) gains the highest profit.

choosing	$U(B)$	$U(B)$ with fee
Random Picker	0.50	0.50
Two Sellers Price Compare	0.51	0.41
All Sellers Price Compare	0.52	0.32
Intelligent Buyer	0.54	0.52

Table 3: All sellers apply MY, and the fee was set to 0.1

We had questions about the profitability of the InfoCenter, since it has to handle the entire load of the buyers' RPQs. The InfoCenter remains profitable, as can be seen in Table 2. First, the InfoCenter can collect a fee in order to get back some of the costs for handling RPQs. Second, the InfoCenter can offer unique information using the InfoSPs' service. In that way, the InfoCenter can sell that information at higher prices, and thus handle higher operational costs. The InfoCenter gains the highest profit when implementing the GT pricing algorithm, which is a fee-free algorithm. We assume that in an e-market with several InfoCenters, the highest profit will be gained with DFwMY and not with GT. We leave confirmation of this to future work.

When we examine how fees affect buyers' profit, on the one hand, we expect that price comparison will increase the buyers' profit. As we can see in Table 3, the price comparison decreases the buyers' profit when fees are charged. This is because the difference in profit is not significant, since the difference in sellers' prices is not significant (due to the price war). As a result, when sellers collect a fee for RPQs, the cost of comparing prices is larger than the increase in profit, therefore buyers earn the highest profit when not comparing prices. The intelligent buyer, on the other hand, compares prices only when it believes that it can benefit from it. Selective price comparison leads the intelligent buyer to earn the highest profit. Furthermore, since sellers do not know when the intelligent buyer will compare prices, they might reasonably consider buyers to always be comparing prices. Thus, sellers would not set fixed prices.

The results show that buyers gain more from not comparing prices when a fee is used. Furthermore, e-markets with additional sellers will increase the cost of comparing prices of all sellers. Therefore, if buyers prefer not to compare prices, then sellers will have no incentive to reduce prices and will set the price to be v. In other words, if buyers have no incentive to compare prices, then sellers have no incentive to reduce prices, which will force buyers to think twice about their decision not to compare prices.

6 Conclusions and Summary

We presented a model for an information e-marketplace that includes InfoCenter agents as intermediaries of information. We extended Yarom et al. work [Yarom *et al.*, 2002] by studying four desiderata for the sellers', buyers' and InfoCenters' utility functions. We defined potential characteristics of the agents' utility functions, and presented why they are important. Furthermore, we defined a utility function that incorporates those characteristics. We continued by showing that maximizing utility is not a trivial task, and trading situations engender behavior such as appears in the prisoner's dilemma.

We focused on markets where requests for price quotes may incur a cost, analyzing the impact that these fees will have on the design of strategic behavior for automated trading agents. We also addressed the problem of handling a large number of request for price quotes (RPQs) by the sellers. RPQs can come from buyers that compare prices in order to find the seller with the lowest price. RPQs can come also from sellers that look at other sellers' prices in order to use this data when setting their own prices. We present two ways to handle this load, first by collecting a fee for each RPQ (buyers that perform transactions get the fee back), and second by using the InfoCenter agent to handle the buyers and the buyers' RPQs.

We showed that a significant reduction in the load of RPQs will occur when sellers delegate the task of handling buyers to the InfoCenters. We found that the InfoCenter, even when it needs to handle large numbers of RPQs, will remain profitable. This is due to the fact that it can collect fees to recoup some of the cost of handling those RPQs. Moreover, it can use its capabilities to introduce attractive information products that will produce higher profits.

The ideal alternative in an e-market with fees on RPQs is to implement intelligent buyers. Intelligent buyers decide when comparing prices is needed, and how many prices to compare. Thus sellers will have an incentive to reduce prices, while intelligent buyers will ask for additional RPQs only when they are needed.

The price behavior in an e-market with fees for RPQs is similar to the behavior in an e-market without fees. The only exception is when none of the buyers compare prices, in which case this will lead to a market with $price$ equal to v (the buyer's value of the information product). The effect of adding an InfoCenter to an e-market does not change the behavior of prices. The behavior of the price of new information (which the InfoCenter manipulates) is as if the InfoCenter were a seller holding that information. In this paper, an e-market with a single InfoCenter was analyzed. In [Yarom *et al.*, 2002], Yarom et al. showed that this price behavior is similar to the one obtained in e-markets with several InfoCenters. As for the price behavior of the information that sellers sell, since the InfoCenter compares prices and resells this information to buyers, price behavior remains similar to the market without an InfoCenter.

References

[Carmel and Markovitch, 1999] David Carmel and Shaul Markovitch. Exploration strategies for model-based learning in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 2(2):141–172, 1999.

[Kephart *et al.*, 2000] J. O. Kephart, J. E. Hanson, and A. R. Greenwald. Dynamic pricing by software agents. *Computer Networks*, 32(6):731–752, May 2000.

[Ketchpel *et al.*, 1997] S. Ketchpel, H. Garcia-Molina, and A. Paepcke. Shopping models: A flexible architecture for information commerce. In *Proceedings of Digital Libraries*, 1997.

[Russell and Wefald, 1991] S. Russell and E. Wefald. Principles of metareasoning. *Artificial Intelligence*, 49:361–395, 1991.

[Yarom *et al.*, 2002] I. Yarom, C. V. Goldman, and J. S. Rosenschein. The impact of InfoCenters on E-marketplaces. In *Proceedings of AAMAS-02*, pages 1290–1291, Bologna, Italy, 2002.