The Impact of Nested Agent Models in an Information Economy

José M. Vidal and Edmund H. Durfee
Artificial Intelligence Laboratory, University of Michigan
1101 Beal Avenue, Ann Arbor, MI 48109-2110
jmvidal@umich.edu

Abstract

We present our approach to the problem of how an agent, within an economic Multi-Agent System, can determine when it should behave strategically (i.e. model the other agents), and when it should act as a simple price-taker. We provide a framework for the incremental implementation of modeling capabilities in agents. These agents were implemented and different populations simulated in order to learn more about their behavior and the merits of using agent models. Our results show, among other lessons, how savvy buyers can avoid being "cheated" by sellers, how price volatility can be used to quantitatively predict the benefits of deeper models, and how specific types of agent populations influence system behavior.

Topic Areas: Agent Modeling/Learning, Economic Societies of Agents.

Introduction

When designing open multi-agent systems (i.e. those that allow anyone to add other agents to the system), one must consider how these agents will interact, and design protocols that discourage agents from spending time and computation trying to take advantage of others and encourage them to achieve actual domain tasks. Unfortunately, this seems to be possible only in very restrictive domains (Rosenschein & Zlotkin 1994). Specifically, it is not possible if we situate our agents in an economic society of agents, such as the University of Michigan Digital Library[1] (Atkins *et al.* 1996)(UMDL). Here the agents will be responsible for making their own decisions about when to buy/sell and who to do business with. Market systems (e.g. auctions) will be implemented as part of the UMDL in order to facilitate the transactions. These mechanisms will, sometimes, diminish the benefits that might come from making "strategic" decisions. However, the individual agent has to decide when to let its own welfare rest in the hands of the market mechanism, and when it should take over and make more strategic decisions.

In this paper we present our approach to the problem of how an agent, within an economic MAS, can determine when it should behave strategically (i.e. model the other agents), and when it should act as a simple price-taker and buy/sell at the lowest price possible. We will show how, in some circumstances, agents benefit by building and using models of others while other times the extra effort is wasted. Our results point to metrics that can be used to make quantitative predictions as to the benefits obtained by using deeper models.

Description of the UMDL

The UMDL project is a large-scale, multidisciplinary effort to design and build a flexible, scalable infrastructure for rendering library services in a digital networked environment. In order to meet these goals, we chose to implement the library as a collection of interacting agents, each specialized to perform a particular task and all of them acting in an artificial economy. These agents buy and sell goods/services from each other in an effort to make a profit. Since the UMDL is an open system, which will allow third-parties to build and integrate their own agents into the architecture, we treat all agents as purely selfish.

Implications of the information economy. Information goods/services, like those provided in the UMDL, are very hard to compartmentalize into equivalence classes that all agents can agree on, especially if the agents are software agents who are constantly changing their preferences in order to maximize their profits. For example, if an encyclopedia database access is defined as a good, then all agents providing encyclopedia accesses can be considered as selling the same good. It is likely, however, that a buyer of this good might decide that seller $s1$ provides better answers than seller $s2$. We cannot possibly hope to enu-

[1] http://www.si.umich.edu/UMDL/HomePage.html

merate the set of reasons an agent might have for preferring one set of answers over another, and we should not try to do so. It should be up to the individual buyers to decide what items belong to the same good category, each buyer clustering items in, possibly, different ways.

This situation is even more evident when we consider an information economy rooted in some information delivery infrastructure (e.g. the Internet). There are two main characteristics that set this economy apart from a traditional economy.

- There is virtually no cost of reproduction. Once the information is created it can be duplicated virtually for free.
- All agents have virtually direct and free access to all other agents.

If these two characteristics are present in an economy, it is useless to talk about supply and demand, since supply is practically infinite for any particular good and available everywhere. The only way agents can survive in such an economy is by providing value-added services that are tailored to meet their customers' needs. Each provider will try to differentiate his goods from everyone else's while each buyer will try to find those suppliers that best meet her value function.

A Simplified Model of the UMDL

In order to capture the main characteristics of the UMDL, and to facilitate the development and testing of agents, we have defined an "abstract" economic model. We define an economic society of agents as one where each agent is either a *buyer* or a *seller*. The set of buyers is B and the set of sellers is S. These agents exchange goods by paying some price $p \in P$, where P is a finite set. The buyers are capable of assessing the quality of a good received and giving it some value $q \in Q$, where Q is also a finite set.

The exchange protocol, seen in Figure 1, works as follows: When a buyer $b \in B$ wants to buy a good g, she will advertise this fact. Each seller $s \in S$ that sells that good will give his bid in the form of a price p_s^g. The buyer will pick one of these and will pay the seller. The seller will then return[2] the specified good. Note that there is no law that forces the seller to return a good of any quality. It is up to the buyer to assess the quality q of the good. Each buyer b also has a value function for

[2]In the case of agent/link failure, each agent is free to set its own timeouts and assess the quality of the never-received good accordingly. Bids that are not received in time will, of course, not be considered.

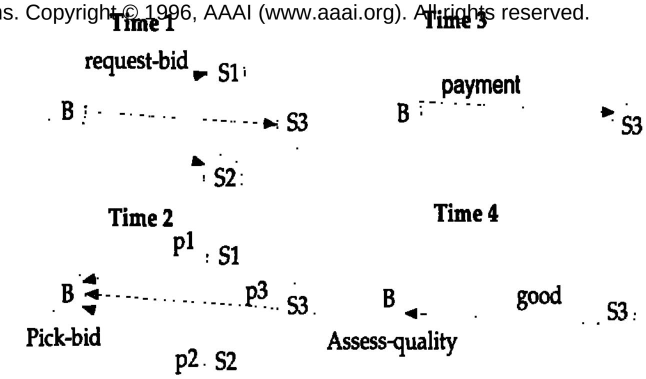

Figure 1: View of the protocol. We show only one buyer B and three sellers $S1$, $S2$, and $S3$. At time 1 the buyer requests bids for some good. At time 2 the sellers send their prices for that good. At time 3 the buyer picks one of the bids, pays the seller the amount and then, at time 4, she receives the good.

each good $g \in G$ that it might wish to buy. The value function, $V_b^g(p, q)$ returns a number that represents the value that b assigns to that particular good at that particular price and quality. Each seller $s \in S$, on the other hand, has a cost c_s^g associated with each good it can produce. Therefore, if seller s gets paid p for good g, his profit will be $\text{Profit}(p, c_s^g)$. Since we assume that cost and payments are expressed in the same unit (i.e. money), the profit equation simplifies to $p - c_s^g$. The buyers, therefore, have the goal of maximizing the value they get for their transactions, and the sellers have the goal of maximizing their profits.

Learning recursive models

Agents placed in the economic society we just described will have to learn, via trial and error, what actions give them the highest expected reward and under which circumstances. In this section we will present techniques that these agents might use to maximize their rewards.

An important question we wish to answer is: when do agents benefit from having deeper (i.e. more complex) models of other agents? It should be intuitive that, ignoring computational costs, the agents with more complete models of others will always do better. This seems to be usually true, however, there are instances when it is significantly better to have deeper models, and instances when the difference is barely noticeable. These instances are defined in part by the set of other agents present and their capabilities and preferences. In order to precisely determine what these instances are, and in the hopes of providing a more general framework for studying the effects of increased agent-modeling capabilities within our economic model, we defined a set of techniques that our agents can use for learning and using models.

We divide the agents into classes that correspond to their modeling capabilities. The hierarchy we present is inspired by RMM (Gmytrasiewicz 1996), but is function-based rather than matrix-based, and includes learning. We start with agents with no models (also referred to as 0-level agents), who must base their actions purely on their inputs and the rewards they receive. They are not aware that there are other agents out there. Agents with 1-level models are aware that there are other agents out there but have no idea what the "interior" of these agents looks like. That is, in RMM terminology, they are incapable of ascribing intentions to others. They must make their predictions simply based on the previous actions of the other agents, by building sub-intentional models of others. Agents with 2-level models have intentional models of others (i.e. have models of their beliefs and inference processes) and believe that others keep sub-intentional (i.e. 1-level) models of others. We can similarly keep defining agents of three, four, five-level models, but so far we have concentrated only on the first three levels. In the following sections, we talk about each one of these in more detail and give details about their implementation. Our current theory only considers agents that are either buyers or sellers, but not both.

Agents with no models

Agents with no models must learn everything they know from observations they make about the environment, and from any rewards they get. In our economic society this means that buyers see the bids they receive and the good received after striking a contract, while sellers see the request for bids and the profit they made (if any). In general, agents get some input, take an action, then receive some reward. This is the same basic framework under which most learning mechanism are presented. We decided to use a form of reinforcement learning (Sutton 1988) (Watkins & Dayan 1992) for implementing this kind of learning in our agents, since it is a simple method and the domain is simple enough for it to do a reasonable job.

Both buyers and sellers will use the equations in the next sections for determining what actions to take. However, with a small probability ϵ they will choose to explore, instead of exploit, and will pick their action at random (except for the fact that sellers never bid below cost). The value of ϵ is initially 1 but decreases with time to some empirically chosen, fixed minimum value $\epsilon_{\min}$. That is,

$$\epsilon_{t+1} = \begin{cases} \gamma\epsilon_t & \text{if } \gamma\epsilon_t > \epsilon_{\min} \\ \epsilon_{\min} & \text{otherwise} \end{cases}$$

where $0 < \gamma < 1$ is some annealing factor.

Buyers with no models. A buyer b will start by requesting bids for a good g. She will then accept all bids for good g and will pick the seller:

$$s^* = \arg_{s\in S} \max f^g(p_s^g) \tag{1}$$

The function $f^g(p)$ returns the value the buyer expects to get if she buys good g at a price of p. It is learned using a simple form of reinforcement learning, namely:

$$f_{t+1}^g(p) = (1-\alpha)f_t^g(p) + \alpha \cdot V_b^g(p,q) \tag{2}$$

where α is the learning rate, p is the price b paid for the good, and q is the quality she ascribed to it. The learning rate is initially set to 1 and, like ϵ, is decreased until it reaches some fixed minimum value $\alpha_{\min}$.

Sellers with no models. When asked for a bid, the seller s will provide one whose price is greater than or equal[3] to the cost for producing it (i.e. $p_s^g \geq c_s^g$). From these prices, he will chose the one with the highest expected profit:

$$p_s^* = \arg_{p\in P} \max h_s^g(p) \tag{3}$$

The function $h_s^g(p)$ returns the profit s expects to get if he offers good g at a price p. It is also learned using reinforcement learning, as follows:

$$h_{t+1}^g(p) = (1-\alpha)h_t^g(p) + \alpha \cdot \text{Profit}_s^g(p) \tag{4}$$

where

$$\text{Profit}_s^g(p) = \begin{cases} p - c_s^g & \text{if he wins auction} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Agents with One-level Models

The next step is for the agents to keep one-level models of the other agents. This means that it has no idea of what the interior (i.e. "mental") processes of the other agents are, but it recognizes the fact that there are other agents out there whose behaviors influence its rewards. The agent, therefore, can only model others by looking at their past behavior and trying to predict, from it, their future actions.

[3]We could just as easily have said that the price must be strictly greater than the cost.

Buyers with one-level models. Each buyer can now keep a history of the qualities it ascribes to the goods returned by each seller. She can, in fact, remember the last N qualities returned by some seller s for some good g, and define a probability density function $q^g_s(x)$ over the qualities x returned by s (i.e. $q^g_s(x)$ returns the probability that s returns an instance of good g that has quality x). She can use the expected value of this function to calculate which seller she expects will give her the highest expected value.

$$\begin{aligned} s^* &= \arg_{s \in S} \max E(V^g_b(p^g_s, q^g_s(x))) \\ &= \arg_{s \in S} \max \frac{1}{|Q|} \sum_{x \in Q} q^g_s(x) \cdot V^g_b(p^g_s, x) \end{aligned} \quad (6)$$

The buyer does not need to model other buyers since they do not affect the value she gets.

Sellers with one-level models. Each seller will try to predict what bid the other sellers will submit (based solely on what they have bid in the past), and what bid the buyer will likely pick. A complete implementation would require the seller to remember past combinations of buyers, bids and results (i.e. who was buying, who bid what, and who won). However, it is unrealistic to expect a seller to remember all this since there are at least $|P|^{|S|} \cdot |B|$ possible combinations.

We believe, however, that the seller's one-level behavior can be approximated by having him remember the last N prices accepted by each buyer b for each good g, and form a probability density function $m^g_b(x)$, which returns the probability that b will accept(pick) price p for good g. Similarly, the seller remembers other sellers' last N bids for good g and forms $n^g_s(y)$, which gives the probability that s will bid y for good g. The seller s can now determine which bid maximizes his expected profits.

$$\begin{aligned} p^* = &\arg_{p \in P} \max(p - c^g_s) \cdot \\ &\prod_{s' \in \{S-s\}} \sum_{p' \in P} \begin{cases} n^g_{s'}(p') & \text{if } m^g_b(p') \le m^g_b(p) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (7)$$

Note that this function also does a small amount of approximation by assuming that s wins whenever there is a tie[4]. The function calculates the best bid by determining, for each possible bid, the product of the profit and the probability that the agent will get that profit. Since the profit for lost bids is 0, we only need to consider the cases where s wins. The probability that s will win can then be found by calculating the product of the probabilities that his bid will beat the bids of each of the other sellers.

[4]The complete solution would have to consider the probabilities that s ties with 1, 2, 3,... other agents. In order to do this we would need to consider all $|P|^{|S|}$ subsets.

Agents with Two-level Models

Two level models consist of an intentional model of the agent being modeled (which contains the agent's desires), and the one-level models that the agent being modeled keeps of the other agents (these form part of the agent's beliefs about others). Our intentional models correspond to the procedures or functions used by agents that use one-level models.

Buyers with two-level models. Since the buyer receives bids from the sellers, there is no need for her to try to out-guess, or predict what the sellers will bid. She is also not concerned with what the other buyers are doing since, in our model, there is an effectively infinite supply of goods. The buyers are, therefore, not competing with each other and do not need to keep deeper models of others.

Sellers with two-level models. He will model other sellers as if they were using the one-level models. That is, he thinks they will model others using policy models and make their decisions using the equations presented in Section . He will try to predict their bids and then try to find a bid for himself that the buyer will prefer more than all the bids of the other sellers. His model of the buyer will also be an intentional model. He will model the buyers as though they were implemented as explained in Section .

The algorithm he follows is to first use his models of the sellers to predict what bids p_i they will submit. He has a model of the buyer $C(s_1 \cdots s_n, p_1 \cdots p_n)$, that tells him which bid she might pick given the set of bids p_i submitted by all sellers s_i. The seller s_j uses this model to determine which of his bids will bring him higher profit, by first finding the set of bids he can make that will win:

$$P' = \{p_j | p_j \in P, C(s_1 \cdots s_j \cdots s_n, p_1 \cdots p_j \cdots p_n) = j\} \quad (8)$$

And from these finding the one with the highest profit:

$$p^* = \arg_{p \in P'} \max(p - c^g_s) \quad (9)$$

Tests

Since there is no obvious way to analytically determine how different populations of agents would interact and, of greater interest to us, how much better (or worse) the agents with deeper models would fare, we decided to implement a society of the agents described above

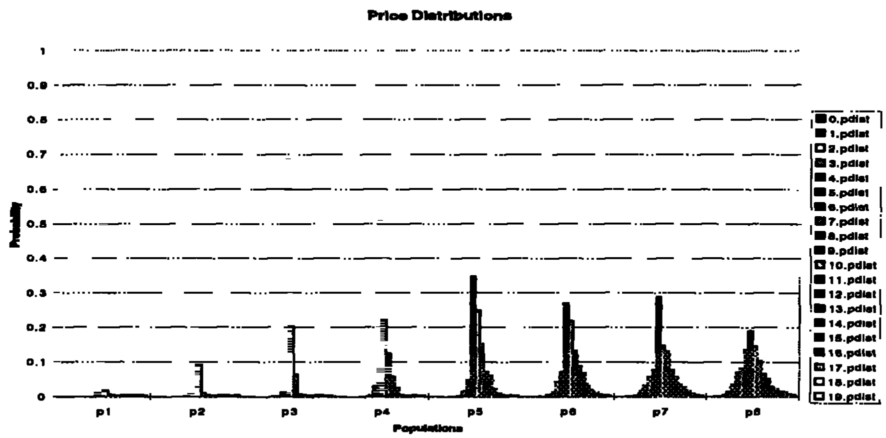

Figure 2: Price distributions for populations of 0-level buyers and sellers. The prices are $0 \cdots 19$. The columns represent the percentage of time the good was sold at each price, in each population. In *p1* sellers return qualities $\{8,8,8,8,8,8,8,8\}$, in *p2* its $\{8,8,8,8,8,8,7,8\}$, and so on such that by *p8* its $\{1,2,3,4,5,6,7,8\}$. The highest peak in all populations corresponds to price 9.

and ran it to test our hypotheses. In all tests, we had 5 buyers and 8 sellers. The buyers had the same value function $V_b(p,q) = 3 \cdot q - p$, which means that if $p = q$ then the buyers will prefer the seller that offers the higher quality. All sellers had costs equal to the quality they returned in order to support the common wisdom assumption that quality goods cost more to produce. We also set $\alpha_{\min} = .1$, $\epsilon_{\min} = .05$, and $\gamma = .99$. There were 100 runs done for each population of agents, each run consisting of 10000 auctions (i.e. iterations of the protocol). The lessons presented here are based on the averages of these 100 runs.

Lessons

Micro versus macro behaviors. In all tests we found the behavior for any particular run does not necessarily reflect the average behavior of the system. The prices have a tendency to get "stuck" into particular patterns, which eventually change. The results presented here, therefore, are based on averages of many runs. While these averages seem very stable, and a good first step in learning to understand these systems, in the future we will need to address micro-level issues. So far, we notice that the micro-level behaviors are much more closely tied, usually in intuitive ways, to the agents' learning rates α and exploration rates ϵ. That is, higher rates (for both) lead to more volatile pricing behavior.

0-level buyers and sellers. This type of population is equivalent to a "blind" auction, where the agents only see the price and the good, but are prevented from seeing who the seller (or buyer) was. As expected, we found that an equilibrium price[5] is reached as long as all the sellers are providing the same quality. Otherwise, if the sellers offer different quality goods, the price fluctuates as the buyers try to find the best price to buy at and the sellers try to find the price[6] the buyers favor. In these populations, the sellers offering the higher quality, at a higher cost, lose money. Meanwhile, sellers offering lower quality, at a lower cost, earn some extra income by selling their low quality goods to buyers that expect, and are paying for, higher quality. When they do this, we found that the mean price actually *increases*, evidently because price acts as a signal for quality and the added uncertainty makes the higher prices more likely to give the buyer a higher value. We see this in Figure 2, where population *p1* has all sellers returning the same quality while in each successive population more agents offer lower quality. The mean price increases in successive populations, but eventually decreases, after enough sellers start returning low quality goods.

0-level buyers and sellers, plus one 1-level seller. In these population sets we explored the advantages that a 1-level seller has over identical 0-level sellers. The advantage was non-existent when all sellers returned the same quality (i.e. when the prices reached

[5] That is, p is an equilibrium price if every seller that can sell at that price (i.e. those whose cost is less than p) does.

[6] Remember, the sellers are constrained to return a fixed quality. They can only change the price they charge.

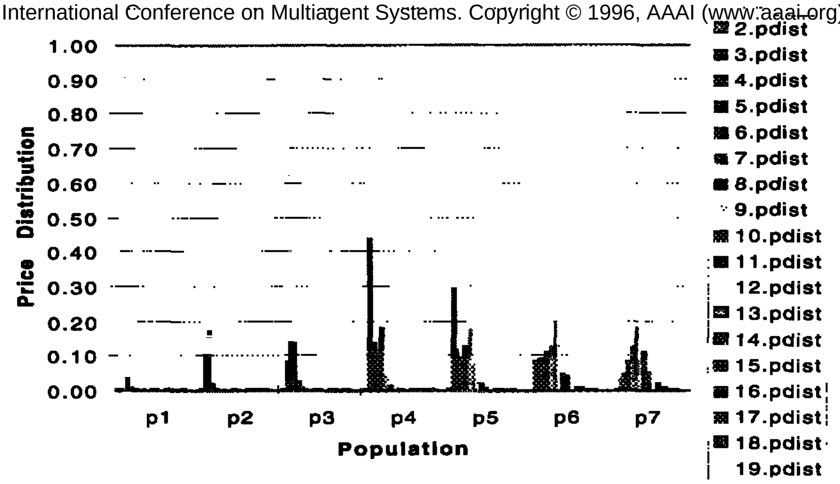

Figure 3: Price distributions for populations of 0-level buyers and 0-level sellers plus one 1-level seller (#12). In *p1* sellers return qualities $\{2,2,2,2,2,2,2,2\}$, in *p2* its $\{2,3,2,2,2,2,2,2\}$, and so on such that by *p7* its $\{2,3,4,5,6,7,8,2\}$. The 1-level seller and seller #5 always return quality 2.

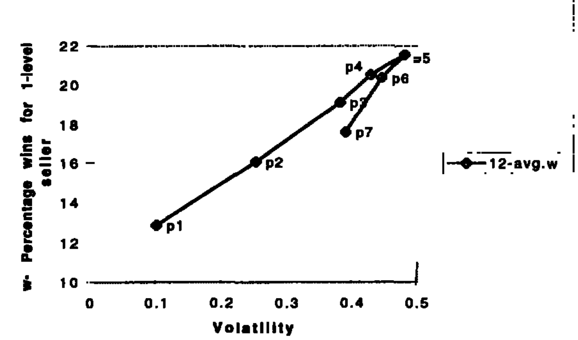

Figure 4: Scatter plot of volatility versus the percentage of time that the 1-level seller wins (w). The populations are the same as in Figure 3.

an equilibrium), but increased as the sellers started to diverge in the quality they returned. In order to make these findings useful when building agents, we needed a way to make quantitative predictions as to the benefits of keeping 1-level models. It turns out that these benefits can be predicted, not by the population type as we had first guessed, but by the price volatility.

We define ***volatility*** as the number of times the price changes from one auction to the next, divided by the total number of auctions. Figure 4 shows the linear relation between volatility and the percentage of times the 1-level seller wins. The two lines correspond to two "types" of volatility. The first line (*p1*-*p5*) is when the buyers' second-favorite (and possibly, the third, fourth, etc) equilibrium price is greater than her most preferred one. In these cases the buyers and sellers fight among the two most preferred prices, the sellers pulling towards the higher equilibrium price, as shown by the two peaks in *p4*-*p5* in Figure 3. The other line correspond to cases where the buyers' preferred equilibrium price is greater than the runner-ups.

The slope of these lines can be easily calculated and the resulting function can be used by a seller agent for making a ***quantitative*** prediction as to how much he would benefit by switching to 1-level models. That is, he could measure price volatility, multiply it by the appropriate slope, and the resulting number would be the percentage of times he would win. However, for this to work the agent needs to know that all eight buyers and five sellers are 0-level modelers. Also, slight changes in our learning parameters ($.02 \leq \epsilon_{\min} \leq .08$ and $.05 \leq \alpha_{\min} \leq .2$) lead to slight changes in the slopes. Still, we hope to eventually generalize this process so that we can make predictions for all possible combinations (i.e. x buyers, y sellers) of agents and learning parameters.

We also want to make clear a small caveat, which is that the volatility that is correlated to the usefulness of keeping 1-level models, is the volatility of the system ***with*** the agent already doing 1-level modeling. Fortunately, having one agent change from 0-level to 1-level does not have a great effect on the volatility as long as there are enough (i.e. more than five or so) other sellers.

In all populations where the buyers are 0-level, we saw that it really pays for the sellers to have low costs because this allows them to lower their prices to fit almost any demand. Since the buyers have 0-level mod-

Buyers	Sellers	Lessons
0-level	0-level	Equilibrium reached only when all sellers offer the same quality. Otherwise, we get fluctuations. Mean price increases when quality offered decreases.
0-level	Any	Sellers have big incentives to lower quality/cost.
0-level	0-level and one 1-level	1-level seller beats others. Quantitative advantage of being 1-level predicted by volatility and price distribution.
1-level	0-level and one 1-level	Buyers have upper hand. They buy from the most preferred seller. 1-level sellers are usually at a disadvantage.
1-level	1-level and one 2-level	Since 2-level have perfect models, they win an overwhelming percentage of time, except when they offer a rather lower quality.

Table 1: Summary of lessons. In all cases the buyers had identical value and quality assessment functions. Sellers were constrained to always return the same quality.

els, these sellers can also raise their prices when appropriate, in effect "pretending" to be the high-quality sellers, and make an even more substantial profit. The buyers are the losers in such situations.

1-level buyers and 0 and 1-level sellers. In these populations the buyers have the upper hand. They quickly identify those sellers that provide the highest quality goods and buy exclusively from them. The sellers do not benefit from having deeper models; in fact, the percentage of times 1-level sellers win is less than that of similar 0-level sellers because the 1-level sellers try to charge higher prices than the 0-level sellers. The 1-level buyers do not fall for this trick– they know what quality to expect, and buy more from the lower-priced 0-level sellers. We have here a case of erroneous models– 1-level sellers assume that buyers are 0-level, and since this is not true, their erroneous strategies lead them to make bad decisions.

As the number of competing (i.e. offering the same quality) sellers increases, the 1-level sellers increase their profits because the 0-level sellers do not win as much (increased competition means everybody wins less) and take longer to converge on the best price. The 1-level seller takes advantage of this time lag but, in the end, he can not do better than 0-level sellers.

1-level buyers and 1 and 2-level sellers. Assuming that the 2-level seller has perfect models of the other agents, we find that he wins an overwhelming percentage of the time. This is true, surprisingly enough, even when some of the 1-level sellers offer slightly higher quality goods. However, when the quality difference becomes too great (i.e. greater than 1), the buyers finally start to buy from the high quality 1-level sellers.

Conclusions

We have presented a framework for the development of agents with incremental modeling/learning capabilities, in an economic society of agents. These agents were built, and the execution of different agent populations lead us to the discovery of the lessons summarized in Table 1. The discovery of volatility and price distributions as predictors of the benefits of deeper models, will be very useful when building agents. Moreover, we believe that other similar metrics will allow deeper modeling agents to make similar predictions. We are also encouraged by the fact that increasing the agents' capabilities changes the system in ways that we can recognize from our everyday economic experience.

Some of the agent capabilities shown in this paper are already being implemented into the UMDL (Atkins *et al.* 1996) MAS. Our results showed how sellers with deeper models fare better, in general, even when they produce less valuable goods. This means that we should expect those type of agents to, eventually, be added into the UMDL[7]. Fortunately, this advantage is diminished by having buyers keep deeper models. We expect that there will be a level at which the gains and costs associated with keeping deeper models balance out for each agent. Our hope is to provide a mechanism for UMDL agents to dynamically determine this cutoff and constantly adjust their behavior to maximize their expected profits given the current system behavior. The lessons in this paper are a significant

[7]If not by us, then by a profit-conscious third party.

step in this direction.

We are considering the expansion of the model with the possible additions of agents that can both buy and sell, and sellers that can return different quality goods. Allowing sellers to change the quality returned to fit the buyer will make them more competitive against 1-level buyers. We are also continuing tests on many different types of agent populations in the hopes of getting a better understanding of how well different agents fare in the different populations.

In the long run, another offshoot of this research could be a better characterization of the type of environments and how they allow/inhibit "cheating" behavior in different agent populations. That is, we saw how, in our economic model, agents are sometimes rewarded for behavior that does not seem to be good for the community as a whole. The rewards, we are finding, start to diminish as the other agents become "smarter". It would be very useful to characterize the type of environments and agent populations that, combined, foster such antisocial behavior (see (Rosenschein & Zlotkin 1994)), especially as interest in multi-agent systems grows.

References

Akerlof, G. A. 1970. The market for 'lemons': Quality uncertainty and the market mechanism. *The Quaterly Journal of Economics* 488–500.

Atkins, D. E.; Birmingham, W. P.; Durfee, E. H.; Glover, E. J.; Mullen, T.; Rundensteiner, E. A.; Soloway, E.; Vidal, J. M.; Wallace, R.; and Wellman, M. P. 1996. Toward inquiry-based education through interacting software agents. *IEEE Computer.* http://www.computer.org/pubs/computer/-dli/r50069/r50069.htm .

Axelrod, R. 1996. The evolution of strategies in the iterated prisoner's dilemma. Cambridge University Press. forthcoming.

Durfee, E. H.; Gmytrasiewicz, P. J.; and Rosenschein, J. S. 1994. The utility of embedded communications and the emergence of protocols. In *Proceedings of the 13th International Distributed Artificial Intelligence Workshop.*

Epstein, J. M., and Axtell, R. L. 1996. *Growing Artifical Societies: Social Science from the Bottom Up.* Brookings Institution. Description at http://-www.brook.edu/pub/books/ARTIFSOC.HTM .

Gmytrasiewicz, P. J. 1996. On reasoning about other agents. In Wooldridge, M.; Müller, J. P.; and Tambe, M., eds., *Intelligent Agents Volume II — Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*, Lecture Notes in Artificial Intelligence. Springer-Verlag.

Mullen, T., and Wellman, M. P. 1996. Some issues in the design of market-oriented agents. In Wooldridge, M.; Müller, J. P.; and Tambe, M., eds., *Intelligent Agents Volume II — Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*, Lecture Notes in Artificial Intelligence. Springer-Verlag.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter.* Cambridge, Massachusetts: The MIT Press.

Russell, S., and Wefald, E. 1991. *Do The Right Thing.* Cambridge, Massachusetts: The MIT Press.

Russell, S. 1995. Rationality and intelligence. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 950–957.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* 3:9–44.

Tambe, M., and Rosenbloom, P. S. 1996. Agent tracking in real-time dynamic environments. In Wooldridge, M.; Müller, J. P.; and Tambe, M., eds., *Intelligent Agents Volume II — Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*, Lecture Notes in Artificial Intelligence. Springer-Verlag.

Vidal, J. M., and Durfee, E. H. 1995. Task planning agents in the UMDL. In *Proceedings of the 1995 Intelligent Agents Workshop.* http://ai.eecs.umich.edu/-people/jmvidal/papers/tpa/tpa.html .

Vidal, J. M., and Durfee, E. H. 1996. Using recursive agent models effectively. In Wooldridge, M.; Müller, J. P.; and Tambe, M., eds., *Intelligent Agents Volume II — Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages (ATAL-95)*, Lecture Notes in Artificial Intelligence. Springer-Verlag. http://ai.eecs.umich.edu/-people/jmvidal/papers/lr-rmm2/lr-rmm2.html .

Watkins, C. J., and Dayan, P. 1992. Q-learning. *Machine Learning* 8:279–292.