An Adaptive Negotiation Model for Agent-Based Electronic Commerce

Mihaela Oprea

Department of Informatics, University of Ploiesti

Bd. Bucuresti Nr 39, Ploiesti, 2000, Romania

mihaela@mail.upg-ploiesti.ro

Abstract: In a multi-agent system the agents try to adapt to the environment by learning or by an evolutionary process, thus doing an anticipation of the interaction with the other agents. The paper presents an adaptive negotiation model that uses a feed-forward artificial neural network as a learning capability to model the other agent negotiation strategy.

Keywords: learning agent, adaptive negotiation, feed forward artificial neural network, agent-based e-commerce.

Biography: Mihaela Oprea received her BSc degree in Computer Science from University Politehnica Bucharest in 1990 and her PhD degree in Computer Science from University of Ploiesti in 1996. Currently, she is an Associate Professor of Artificial Intelligence and Expert Systems at the University of Ploiesti, Department of Informatics. Her current research interests include applications of multi-agent systems, development of adaptive models for agent-based e-commerce negotiation, applications of expert systems and artificial intelligence techniques in different domains such as environmental protection and optimal control. She has published more than 40 papers in the area of artificial intelligence in different journals and in the proceedings of national and international conferences and workshops. Dr. Oprea is a member of the Romanian Society for Informatics and Automatics, the Slovenian Artificial Intelligence Society, and the IASTED Technical Committee on Artificial Intelligence and Expert Systems.

Introduction

Adaptability and embodiment are two important issues that need to be addressed when designing flexible multi-agent systems. Adaptability allows the generation of a model of the selection process within the system and thus results in internal representations that can indicate future successful interactions [1], [2]. For a software agent its "body" reflects the history of its adaptation in a virtual environment, including the constraints of the environment and the interactions and methods of dealing with the environment. In the context of a multi-agent system, the two properties, adaptability and embodiment, are tightly related to each other. The main purpose of adaptability in the case of a negotiation scenario is to improve the negotiation competence of the agents, based on learning from their interactions with other agents. In other words, to obtain an agreement and eventually, a better or even, the best deal. In this sense it is well known that multi-agent learning can be viewed as a mean to reach equilibrium. In the recent years several solutions were given to software agent adaptability implementation. Among these, the most used in the case of an agent-based negotiation are Q-learning and Bayesian learning. In this paper we propose the solution of a feed forward artificial neural network that can be used by a seller/buyer agent to model the other agent negotiation strategy.

The paper is structured as follows. In section 2 we describe the learning problem in the context of an agent-based negotiation system and we present some solutions adopted so far. In section 3 we describe the adaptive negotiation model that is proposed by the paper. Some preliminary experimental results are discussed in section 4. Finally, in section 5 we conclude the paper, highlighting the future work.

2. Learning in Agent-Based Negotiation

As stated in [3] and [4] negotiation is a general process involving communication between two or more agents that attempts to find a way of mutually satisfying the separate agent's requirements by undertaking appropriate actions. Usually, negotiation involves multiple rounds of exchanging proposals and counter-proposals. In this context it is important to reach the agreement more efficiently, the solution being the use of an adaptive negotiation model [5]. Learning in a multi-agent environment is complicated by the fact that, as other agents learn, the environment effectively changes [6]. When agents are acting and learning simultaneously, their decisions affect and limit what they subsequently learn.

2.1 The Negotiation Strategy Learning Problem

In a multi-agent system the agents must coordinate their activities or pursuits and when conflicts arise they must be able to negotiate. Conflicts may result from simple limited resource contention to more complex issues where the agents disagree because of discrepancies between their domain expertise. Negotiation involves determining a contract under certain terms and conditions. From the perspective of CBB (Consumer Buyer

Behaviour), the negotiation stage is where the price or other terms of the transaction are determined. For example, the negotiation used in commerce include stock markets, fine art auction houses, flower auctions, and various ad-hoc haggling (automobile dealership and commission-based electronic stores). As a coordination technique, negotiation enable to avoid deadlocks and livelocks in multi-agent systems. Negotiation is achieved through the exchange of proposals between agents. In this paper we shall limit the discussion to bilateral negotiation. The negotiation can be made over a set of issues and a proposal consists of a value for each of those issues and is autonomously generated by the agent's strategy. A way to enhance agent's autonomy in a dynamic environment such as an electronic market is to endow its architecture with learning capabilities.

Suppose we have two agents *a* and *b* that are negotiating over a set of issues that are represented by vector X. The negotiation thread at time *t* is the following: $X_{a\leftrightarrow b}^{t} = \{X_{a\rightarrow b}^{1}, X_{b\rightarrow a}^{2}, \ldots, X_{a\rightarrow b}^{t}\}$. $X_{a\rightarrow b}^{t}[j]$ is the proposal made by agent *a* to agent *b* at time *t* for the issue *j* that is under negotiation. Each agent has a utility function U^i, $U^i: S^i \rightarrow \mathbf{R}$. S^i is the state space relevant to agent *i* and **R** is the set of real numbers. The utility function order the states by preference. Each agent has some private and some shared knowledge. For example, each agent knows only its own utility function and can observe the other agent's actions, i.e. proposals. Also, each agent has some beliefs about the state that would result from performing its available actions. An agent can learn from observation of the interactions with the other agent. The proposals made by an agent are generated by his negotiation strategy that is specific to each issue under negotiation. The negotiation strategy is private to each agent. Let's consider the case of two agents, a seller and a buyer, that are negotiating the price of a specific product. Each agent knows its own reservation price, RP, which is how much the agent is willing to pay or to receive in the deal. A possible deal can be made only if there exists an overlapping zone between the reservation prices of the two agents. The agents don't even know if there is an agreement zone. The ideal rule to find if there is an agreement zone is the following:

if $RP_{seller} \leq RP_{buyer}$ **then**

* The zone of agreement exists
* Possible deal

else

* The zone of agreement doesn't exist
* No deal

where RP_{buyer} is the maximum price the buyer is willing to pay for the product, and RP_{seller} is the minimum price the seller will accept to receive. The way in which each agent is making a proposal for the price of the product is named pricing strategy. In figure 1 we present the buyer-seller interaction from the viewpoint of the seller. Usually, it cannot be forecast if an agreement will be made between two agents that negotiate, so a learning capability would help to higher the percentage of deals that can be made in an electronic market. A learning agent makes his decision based on his own reservation price, on his a priori domain knowledge, and on the history of past price proposals that were exchanged. Through learning the agent will form a more accurate model of the other agent and could create some nested agent models [7].

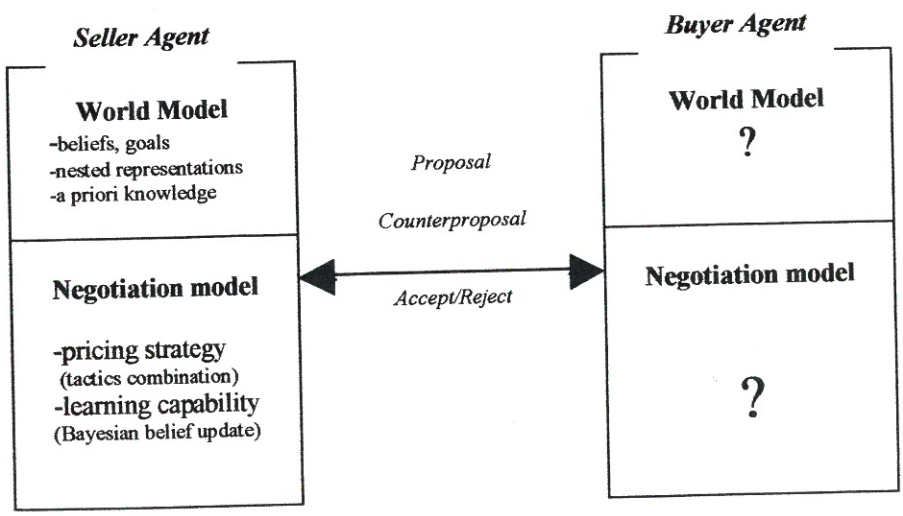

Figure 1: Buyer agent - Seller agent Interaction (from the Viewpoint of the Seller Agent)

2.2 Solutions

In making his decision, a rational agent must take into account the probable choices of others, whose choices are in turn contingent upon his own. This leads to the well known *outguessing regress problem*, where no accurate prediction or confident expectation about the individual choices can be produced. Therefore, all the negotiation models try to avoid this dilemma involved in strategic negotiation [8]. A solution is to enrich the negotiation model by integrating AI planning, Case-Based Reasoning and other decision theoretic techniques. Another solution is to include a learning capability inside the negotiation model that can be combined with decision theoretic techniques. The main solutions that were adopted for the inclusion of a learning capability into a multi-agent system are the reinforcement learning with a special emphasize on Q-learning, Bayesian learning and model-based learning.

Reinforcement learning

The reinforcement learning is a common technique used by adaptive agents in multi-agent systems and its basic idea is to revise beliefs and strategies based on the success or failure of observed performance.

Q-learning is a particular reinforcement learning algorithm (an incremental reinforcement learning) that works by estimating the values of all state-action pairs. An agent that uses a Q-learning algorithm selects an action based on the action-value function, called the Q-function. $Q_j(s, a) = Q_j(s, a) + \lambda(r_j - Q_j(s, a))$, where λ is a constant, r_j is the immediate reward received by agent j after performing action a in state s. The Q-function defines the expected sum of the discounted reward attained by executing an action a in state s and determining the subsequent actions by the current policy π. The Q-function is updated using the agent's experience. The reinforcement learning techniques have to deal with the exploration-exploitation dilemma. Some experimental comparisons between several explore/exploit strategies are presented in [9] showing the risk of exploration in multi-agent systems. In [10] it is demonstrated that genetic algorithm based classifier systems can be used effectively to achieve near-optimal solutions more quickly than Q-learning, this result revealing the problem of slow convergence that is specific to reinforcement learning techniques.

Bayesian learning

Usually, Bayesian behaviour is considered as the only rational agent's behaviour, i.e. the behaviour that maximizes the utility. Bayesian learning is built on bayesian reasoning which provides a probabilistic approach to inference. The bayesian learning algorithms manipulates probabilities together with observed data. In [11] it is presented a sequential decision making model of negotiation called Bazaar, in which learning is modeled as a Bayesian belief update process. During negotiation, the agents use the Bayesian framework to update knowledge and belief that they have about the other agents and the environment. For example, an agent (buyer/seller) could update his belief about the reservation price of the other agent (seller/buyer) based on his interactions with the seller/buyer and on his domain knowledge. The agent's belief is represented as a set of hypotheses. Each agent tries to model the others in a recursive way during the negotiation process, and any change in the environment, if relevant and perceived by an agent, will have an impact on the agent's subsequent decision making. The experiments showed that the greater the zone of agreement, the better the learning agents seize the opportunity.

The Bayesian framework is used as the underlying belief updating mechanism. In addition to providing efficient updating techniques, Bayesian belief networks offer an expressive modeling language and allow easy and flexible encoding of domain-specific knowledge.

Model-based learning

In [12] it is described a model-based learning framework that model the interaction between agents by the game-theoretic concept of repeated games. The approach tries to reduce the number of interaction examples needed for adaptation, by investing more computational resources in deeper analysis of past interaction experience. The learning process has two stages: (1) the learning agent infers a model of the other agent based on past interaction and (2) the learning agent uses the learned model for designing effective interaction strategy for the future. The experimental results presented in [12] showed that a model-based learning agent performed significantly better than a Q-learning agent.

In a model-based learning the interaction between the agents can be viewed as a repeated game from which the agents learn models of the rival agents for exploitation in future encounters. The strategies could be represented by production rules or by any other representation scheme.

3. The Adaptive Negotiation Model

We have extended the service-oriented negotiation model described in [13] with an adaptability component that is implemented by a feed forward artificial neural network. This negotiation model is based on a variation of two parties, many issues value scoring system presented in [14] and was used by generic negotiating agents for business process management applications.

Figure 2 presents the protocol of the service-oriented negotiation model. A protocol of interaction is required because sub-problems interact during domain problem solving and agents have to communicate and interact. The interaction is modelled as an alternating sequences of offers and counteroffers which terminates either with a commitment by both parties to a mutually agreed solution or terminates unsuccessfully. We give a brief description of the negotiation protocol. The two agents, the buyer and the seller, will be referred as the client and the server. Negotiation is initiated when a client utters can_do (transition 1→2). The server can then either indicate that it is capable (2→3) or that it is not (2→failure). If the server is capable, the client may send out a proposal (3→4). The server can reject the proposal (4→failure), accept the proposal (4→5) or counterpropose (4→6). If the server accepts, the client may either deny the contract to the server (5→failure) or confirm the contract (5→success). If the server has counterproposed (4→6) then the client may either accept the new contract (6→7), reject it (6→failure) or counterpropose a new contract (6→4). Between states 4 and 6 there may be several transitions. If the client accepts the contract (6→7), then the server may decide to either award the contract to the client (7→success) or deny it to that client (7→failure).

Negotiation can range over a number of quantitative (e.g. price, cost, duration) and qualitative (e.g. type of reporting policy, nature of the contract) issues. Quantitative issues in negotiation are defined over a real domain (i.e. $x[j] \in D_j = [min_j, max_j]$). Qualitative issues are defined over a totally ordered domain (i.e. $x[j] \in D_j = \langle q_1, \ldots, q_n \rangle$). When an agent receives an offer $X = (x[1], x[2], \ldots, x[n])$, where n is the total number of issues, it rates it by using a function that combines the scores of the different issues (by a linear combination):

$$V^a(x^t) = \sum_{1 \le j \le n} w_j^a(t) V_j^a(x^t[j])$$

where $w_j^a(t)$ is the importance of issue j for agent a at time t.

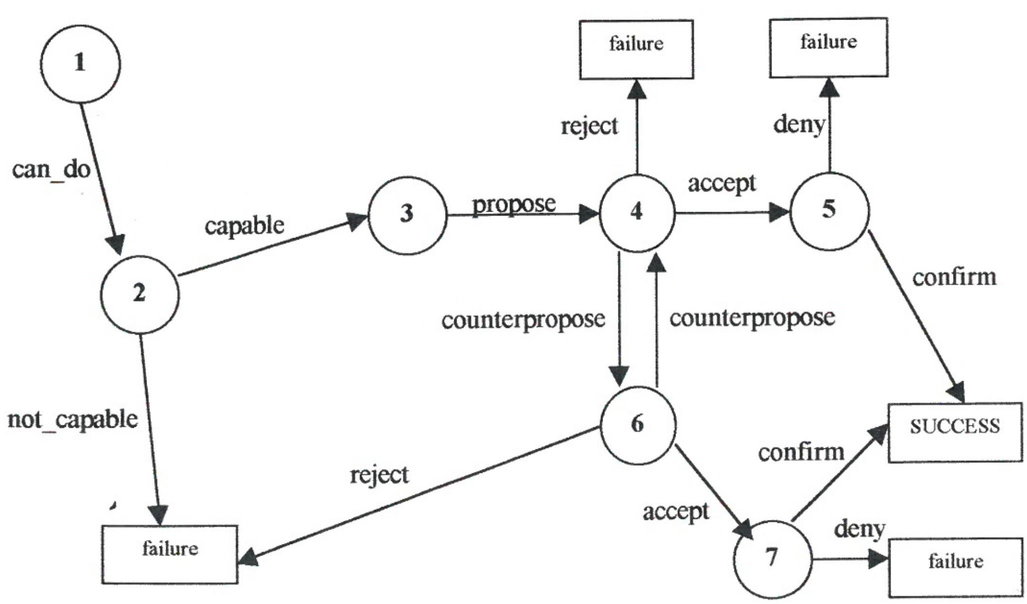

Figure 2: The Negotiation Protocol in the Service-Oriented Model

Each agent has a scoring function $V_j^a : D_j^a \rightarrow [0, 1]$ that gives the score agent a assigns to a value of issue j in the set of its acceptable values D_j^a . For quantitative issues the scoring functions are monotonous. If the score of the received offer is greater than the score of the counter offer the agent would send at this point, then the offer is accepted. If the preset constant deadline (t_{max}^a) at which the negotiation must have been completed by agent a is reached, the offer is rejected by a. Otherwise, a counter offer is sent. Figure 3 describes the price negotiation context.

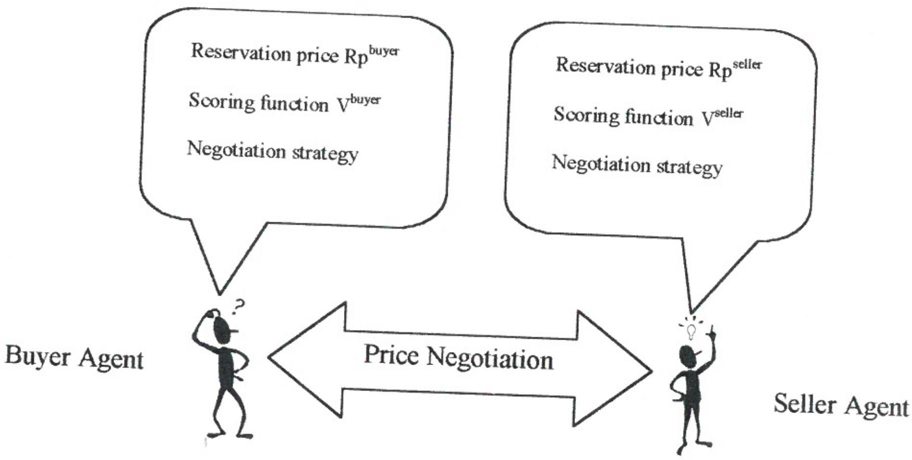

Figure 3: The Price Negotiation Between two Agents – a Seller and a Buyer

If both negotiators use an additive scoring function, Raiffa showed that it is possible to compute the optimum value of X (the contract) as an element on the efficient frontier of negotiation. It has been demonstrated that negotiation convergence is achieved when the scoring value of the received offer is greater than the scoring value of the counter-offer the agent is intended to respond with. The aim of agent i's negotiation strategy is to determine the best course of action which result in an agreement on a contract X that maximizes its scoring function V^a. Each agent has a set of tactics that are used to build negotiation strategies. A new counter-offer is a combination of tactics and how this combination is made will be decided by the negotiation agent's strategy.

In this service-oriented negotiation model several negotiation tactics could be applied in order to determine how to compute the value of an issue (price, volume, quality etc) by considering a single criterion (e.g. time, resources). Some time dependent tactics are Boulware and Conceder. While Boulware tactic maintains the offered value until the time is almost exhausted, whereupon it concedes up to the reservation value, the Conceder tactic goes quickly to the reservation value. Resource dependent tactics are similar to time dependent tactics and they model bounded rationality. Two types of such tactics are dynamic deadline tactics and resource estimation tactics. Finally, the behaviour-dependent tactics are relative Tit-For-Tat (the agent reproduces in percentage terms the behaviour that its opponent performed $\delta \geq 1$ steps ago – if $n > 2\delta$), Random Absolute Tit-For-Tat (same as relative Tit-For-Tay but in absolute terms) and Averaged Tit-For-Tat (the agent computes the average of percentages of changes in a window of size $\gamma \geq 1$ of its oponents history when determining its new offer – if $n > 2\gamma$).

If decision makers could take into consideration what the other agents are thinking and are planning to do, and so learn during their interactions how other agents behave, their payoff might increase. In this order of ideas we have used a connectionist method to implement the learning capability into the negotiation model. The approach was adopted starting from the observation that the exchanged proposals are some time series for each issue under negotiation, and as feed forward artificial neural networks has proved to be good predictors we have included such a neural network in the negotiation model of a learning agent. For simplicity, we shall limit our discussion to an agent-based e-commerce negotiation of a single issue, the price of a specific product, and we shall discuss only about the learning capability of a seller agent, the case of a buyer agent being simmetrical. The history of past price proposals exchanged between a buyer and a seller is given by:

$$X_{b\leftrightarrow s}^{t} = \{ X_{b\rightarrow s}^{1}, X_{s\rightarrow b}^{2}, X_{b\rightarrow s}^{3}, \ldots, X_{b\rightarrow s}^{t-2}, X_{s\rightarrow b}^{t-1}, X_{b\rightarrow s}^{t} \}.$$

This history of price proposals is a time series and so the prediction of the next price proposal can be made by using a feed forward artificial neural network. Suppose that the seller agent has implemented the learning capability as a feed forward neural network. At time t (t>5), the seller has to decide the next price proposal of the buyer based on the past three buyer price proposals. The architecture of the neural network is 3×n×1 (see figure 4). The number of nodes in the hidden layer is set during the network training. The inputs set is Inp = $\{X_{b\to s}^{t-5}, X_{b\to s}^{t-3}, X_{b\to s}^{t-1}\}$, t>5. The output is $X_{b\to s}^{t+1}$. Based on the predicted buyer price proposal $X_{b\to s}^{t+1}$, the seller could become more flexible in order to make a deal more quickly and as convinient as possible from his utility function viewpoint. Sometimes a suboptimal solution could be accepted if a deal can be made. The learning capability is activated during the process of proposals and counterproposals exchanging and will influence the way in which the negotiation will evolve to an agreement.

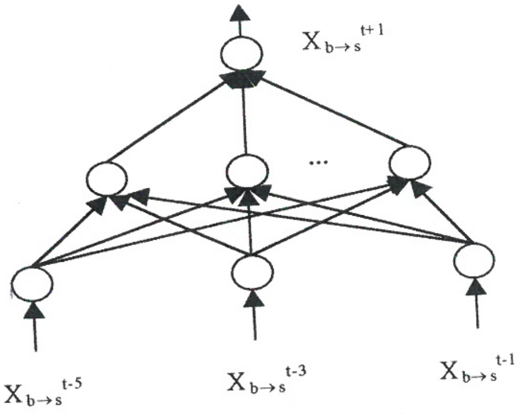

Figure 4: The Artificial Neural Network

Suppose at time *t* the seller will forecast the next price proposal of the buyer, $X_{b\to s}^{t+1}$, and based on this value and on the forecast of the buyer negotiation strategy, will decide which strategy to adopt. The core rule of the decision algorithm is the following:

If V_j^a $(X_{b\to s}^{t+1}[j]) \geq V_j^a$ $(X_{s\to b}^{t}[j])$ **then**

* proposal of the seller is $X_{b\to s}^{t+1}[j] + \varepsilon$

else

* proposal of the seller is $X_{s\to b}^{t}[j] - \varepsilon$

where *j* is the price issue, *a* is the seller agent, ε is a domain-dependent value, and the seller proposal is computed according to one of its tactics depending on the corresponding buyer strategy forecast.

$\varepsilon = \mathrm{trunc}\,(\,|\,X_{s\to b}^{t}[j] - X_{b\to s}^{t+1}[j]\,|\,/\,d\,)$, where d is the division number, $d \in \{2, 3, 4, 5\}$.

Therefore, the seller try to adjust its negotiation strategy to the buyer's negotiation strategy by using the artificial neural network forecasts. The learning capability of the seller agent will give information about the overall negotiation style (e.g. the other agent is tough or compliant) and will provide some heuristic information regarding the buyer's reservation price. As a result of learning, the seller is expected to gain more accurate expectation of the buyer's payoff structure and therefore make more advantageous offers. The domain-specific knowledge could also help in order to make a better estimation of the other agent's reservation price. For example, in some businesses people will offer a price which is above their reservation price by a certain percentage, x%.

A very important observation that we have to make is that the proposed approach could be useful when there is plenty of time for negotiation, i.e. for medium and long term deadlines. The neural network will be adapted to different negotiation contexts. Because the training set is extracted on line, during negotiation, at the beginning of each negotiation, there will be a time window in which the neural network should adapt to the new context and so the learning capability cannot be exploited by the negotiation strategy during the first p proposals ($p \leq 5$). In order to keep the architecture of the neural network as simple as possible and also, to take into account the current behaviour of the buyer agent and to detect any change in his pricing strategy we have chosen a time window of size three, taken into account the last three buyer price proposals.

During negotiation, the seller can make predictions also about the reservation price of the buyer agent, by observing the price proposals that the buyer made. Taking into account the predicted reservation price of the buyer and the predicted next buyer price proposal, the seller could lower or raise its price over time by choosing a specific negotiation strategy. So, the seller agent will reason about the buyer agent based solely on his observations of buyer's actions. The model that the seller will build about the buyer agent represent the unsure knowledge [15].

4. Preliminary Experimental Results

As a testbed of the connectionist approach we have used an e-commerce market simulation (implemented in Jade) in which the price can be negotiated [16]. We have set experimentally the architecture of the neural network to three input nodes, three hidden nodes and one output node. The seller uses our adaptive negotiation model, while the buyer uses one of three simple negotiation strategies: anxious, cool-headed and frugal (i.e. linear, quadratic and exponential functions). The learning rate of the neural network was variable (some values were 0.01, 0.1, 0.2). The weights initializations were in the range of [-0.1, 0.1]. Initially, ten different runs with different weight initializations were performed. Than, we made 20 runs with two different weight initializations and 10 different training sets. The initialization that gave the best result and that which was the closest to the average performance were selected as the two initializations in the actual 20-time cross-evaluation runs. After the pre-training step, we have used the neural network in different negotiation contexts to make the other training step and than we have used the trained network in the testing step.

The order of who begins the negotiation process is randomly selected (the agent which opens the negotiation process fairs better, irrespective of whether the agent is a buyer or a seller). The e-market was composed of two sellers (one that has the proposed adaptive negotiation model – ANN-agent, and one that has a Q-learning capability – Q-agent) and a number of buyers, randomly included in the e-market. We made a comparison between the performances of the two seller agents. The results are summarized in table 1. In all the experiments the zone of agreement was chosen as being not empty.

Table 1

	Type 1		Type 2		Type 3	
	JU	#p	JU	#p	JU	#p
ANN-agent	0.19	52	0.15	41	0.22	62
Q-agent	0.20	57	0.17	44	0.20	53

The three types of buyer agent's negotiation strategy are: type 1 – frugal, type 2 – anxious, type 3 – cool-headed. We have adopted the same joint utility as in [11] which is defined as:

$(P^{*}-RP_{seller})\times(RP_{buyer}-P^{*})/(RP_{buyer}-RP_{seller})^{2}$ where P^{*} is the agreed price.

In table 1 we give the joint utility (JU) and the number of exchanged proposals (#p). The best performance is reached by the ANN-agent in the case of a cool-headed negotiation strategy of the buyer agent.

Also, we had run several simulations for short and medium-term deadlines with the following time limits (in seconds) of the e-market operation: 300s, 600s, 1800s, 3600s and 18000s and we have counted the number of deals made by the seller agents. The experimental results are summarized in table 2. In all situations, both agents, ANN-agent and Q-agent, were capable to make deals, the greatest number of deals being made by the ANN-agent (32 / 57, in comparison with 24 / 48 for the Q-agent) for the last two intervals of time (i.e. 3600s and 18000s), as expected. For the first interval, the Q-agent made a greater number of deals (4, in comparison with 1, made by the ANN-agent), while for the second and the third, both agents made a similar number of deals (Q-agent – 8 / 16, ANN-agent 7 / 14). All the experimental results are related to the simulations data while in a real e-market things could be quite different, the next step of our work being the use of the ANN-agent in a real world e-market.

Table 2

t (s) / seller	300	600	1800	3600	18000
ANN-agent	1	7	14	32	57
Q-agent	4	8	16	24	48

As the most successful tactics reported in the literature for long term deadlines are *linear*, *patient* and *steady*, we have included this set of tactics in the adaptive negotiation model of the ANN-seller agent. The linear tactic is a time-dependent tactic (see relation (1) with β=1), while the patient and the steady tactics are resource-dependent tactics. All these tactics concede at a steady rate throughout the negotiation process.

$$x^t_{s\to b}[j]=\begin{cases}\min^s_j+\alpha^s_j(t)(\max^s_j-\min^s_j), & \mathrm{V}^s_j \text{ decreasing}\\ \min^s_j+(1-\alpha^s_j(t))(\max^s_j-\min^s_j), & \mathrm{V}^s_j \text{ increasing}\end{cases} \qquad (1)$$

where

$$\alpha^s_j=\left(\frac{t}{t^s_{\max}}\right)^{\frac{1}{\beta}}$$

The ANN-seller agent chooses a tactic from its set of tactics based on the predicted value of the buyer proposal at time t+1, according to the core decision rule of the adaptive negotiation model, explained in section 3.

5. Conclusion and Future Work

The main purpose of the work described in this paper was to study the usefulness of a connectionist approach as a learning capability implementation for a negotiating agent in the context of agent-based electronic commerce. The use of a feed forward artificial neural network as a learning ability of a negotiation model in the context of agent-based e-commerce is new.

The preliminary experimental results obtained so far showed a good behaviour of the neural network in the case of medium and long-term deadlines. The e-markets that seems to be appropriate for our ANN-model are second-hand products selling (cars, computers etc), properties selling and so on. In such cases the negotiation time could be extended to a larger interval, in order to make a better deal.

As a future work we shall include the adaptive negotiation model in a real world e-market, for second hand computers selling and we shall test its performances in different negotiation scenarios.

REFERENCES

1. OPREA M., **Adaptability and Embodiment in Agent-Based E-Commerce Negotiation**, Proceedings of the Workshop Adaptability and Embodiment Using Multi-Agent Systems - AEMAS0*1*, Prague, Czech Republic, 2001, pp. 257-265.

2. OPREA, M., **Adaptability in Agent-Based E-Commerce Negotiation**, tutorial notes of the 20th IASTED International Conference Applied Informatics AI'02 – symposium Artificial Intelligence Applications–AIA'02, February 2002, Innsbruck, Austria.

3. FISHER M., **Characterising Simple Negotiation as Distributed Agent-Based Theorem-Proving – A Preliminary Report**, Proceedings of ICMAS00, Boston, 2000, pp. 127-134.

4. WOOLDRIDGE M., **The theory of multi-agent systems**, lecture notes, UPC-Barcelona, 2000.

5. WOOLDRIDGE M., JENNINGS N., Intelligent agent: theory and practice, **The Knowledge Engineering Review**, 10:2, 1995, pp. 115-152.

6. SEN S., SEKARAN M., Multiagent coordination with learning classifier systems, **Proceedings of IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems**, 1995.

7. VIDAL J., DURFEE E., Learning nested agent models in an information economy, **Journal of Experimental & Theoretical Artificial Intelligence**, 10:3 (1998) pp.291-308.

8. SANDHOLM T., LESSER V., Issues in automated negotiation and electronic commerce: Extending the contract net framework. **In Proceedings of the 1st Int. Conf. on Multi-Agent Systems (ICMAS)**, San Francisco, CA, 1995, pp.328-335.

9. PÉREZ-URIBE A., HIRSBRUNNER B., The Risk of Exploration in Multi-Agent Learning Systems: A Case Study, **Proceedings of the Agents-00/ECML-00 workshop on Learning Agents**, Barcelona, 2000, pp.33-37.

10. SEN S., SEKARAN M., Individual Learning of coordination knowledge, **Journal of Experimental & Theoretical Artificial Intelligence**, 10:3, (1998) pp. 333-356.

11. ZENG D., SYCARA K., How Can an Agent Learn to Negotiate, **Intelligent Agents III. Agent Theories, Architectures and Languages,** LNAI 1193, Springer, 1997, p. 233-244.

12. CARMEL D., MARKOVITCH S., **Model-based learning of interaction strategies in multi-agent systems**, JETAI 10 (1998), pp. 309-332.

13. FARATIN P., SIERRA C., JENNINGS N., Negotiation decision functions for autonomous agents, **Robotics and Autonomous Systems**, 24 (1998), pp. 159-182.

14. RAIFFA H., **The Art and Science of Negotiation**, Harvard University Press, Cambridge, MA, 1982.

15. CHRISMAN L., CARUANA R., CARRIKER W., Intelligent Agent Design Issues: Internal Agent State and Incomplete Perception, in **Proceedings of AAAI Fall Symposium Series: Sensory Aspects of Robotic Intelligence**, Monterey, CA, Nov 15-17, 1991.

16. OPREA M., **Use of the Connectionist Approach to Model an Adaptive Agent-Based E-Commerce Negotiation**. research report, October 2001, University of Ploiesti.