A MAS-BASED NEGOTIATION MECHANISM TO DEAL WITH SATURATED CONDITIONS IN DISTRIBUTED ENVIRONMENTS

Mauricio Paletta
Centro Inv. Inf. y Tec. Comp. (CITEC), Universidad de Guayana (UNEG), Av. Atlántico, Ciudad Guayana 8050, Venezuela

Pilar Herrero
Facultad de Informática, Universidad Politécnica de Madrid (UPM), Campus de Montegancedo S/N; 28660, Madrid, Spain

Keywords: Negotiation, Distributed environment, Multi-agent system, Awareness, Artificial neural network.

Abstract: In Collaborative Distributed Environments (CDEs) based on Multi-Agent System (MAS), agents collaborate with each other aiming to achieve a common goal. However, depending on several aspects, like for example the number of nodes in the CDE, the environment condition could be saturated / overloaded making it difficult for agents who are requesting the cooperation of others to carry out its tasks. To deal with this problem, the MAS-based solution should have an appropriate negotiation mechanism between agents. Appropriate means to be efficient in terms of the time involved in the entire process and, of course, that the negotiation is successful. This paper focuses on this problem by presenting a negotiation mechanism (algorithm and protocol) designed to be used in CDEs by means of multi-agent architecture and the awareness concept. This research makes use of a heuristic strategy in order to improve the effectiveness of agents' communication resources and therefore improve collaboration in these environments.

1 INTRODUCTION

Collaborative Distributed Environments (CDEs) are those in which multiple users in remote locations, usually agents, participate in shared activities aiming to achieve a common goal. The success of achieving this goal in a suitable time (efficiency) and/or to obtain the higher quality of results (effectiveness) in these dynamic and distributed environments depends on implementing an appropriate collaboration by means of the most suitable mechanism. Moreover, this appropriate collaboration mechanism should include a negotiation technique between agents to be used when CDE is saturated. In this paper, saturated means that no node is available to collaborate on a specific need for any other node in the CDE.

Negotiation techniques are used to overcome conflicts, and to make agents come to an agreement instead of persuading each other to accept an established solution (Lin et al, 06). In fact, the importance of negotiation in Multi-Agent Systems (MASs) is likely to increase due to the growth of fast and inexpensive standardized communication infrastructures, which allow separately, designed agents to interact in an open and real-time environment and carry out transactions securely (Wooldridge, 02).

In order to improve time of answer (efficiency), one of the most important aspects related with negotiation between agents is to decide with whom to negotiate. The more fitting the candidate to negotiate is, the faster the agent that requires collaboration can achieve positive results of negotiation. Therefore, the negotiation mechanism should be endowed with an algorithm that will decide with which node in the CDE to negotiate with. Moreover, this algorithm must be able to make a decision based on the current situation and making use of the experience acquired from previous negotiations. Heuristic techniques are a good alternative to achieve this goal.

By using Vector Quantization (VQ) techniques (Kohonen et al, 84), (Makhoul et al, 85), (Nasrabadi et al, 88-1), (Nasrabadi et al, 88-2), (Naylor et al, 88), this paper presents a novel negotiation

Paletta M. and Herrero P. (2010).
A MAS-BASED NEGOTIATION MECHANISM TO DEAL WITH SATURATED CONDITIONS IN DISTRIBUTED ENVIRONMENTS.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Agents*, pages 159-164
DOI: 10.5220/0002708201590164

mechanism for CDEs endowed with a non-supervised Artificial Neural Network (ANN) to decide the most suitable candidate with whom to negotiate. This strategy, based on a Neural-Gas network (NGAS) (Martinetz et al, 91), takes into account the information of *awareness* collaborations occurring in the environment under saturated conditions for achieving the most appropriate future *awareness* situations.

The remainder of this paper is organized as follows. Some background aspects are showed in section 2. Section 3 describes the complete MAS-based negotiation mechanism proposed in this paper. Results of the evaluation of the method are showed in Section 4. Some related work is given in Section 5. Finally, the last section includes the conclusions and outgoing future research related to this work.

2 BACKGROUND

This Section presents some background related with: 1) vector quantization and neural-Gas network; and 2) the collaborative mechanism where the negotiation process presented in this paper is used.

2.1 Vector quantization and NGAS

Vector Quantization (VQ) is the process of quantizing n-dimensional input vectors to a limited set of n-dimensional output vectors referred to as *code-vectors*. The set of possible *code-vectors* is called the *codebook*. The *codebook* is usually generated by clustering a given set of training vectors (called *training set*). Clustering can be described then, as the process of organizing the *codebook* into groups whose members share similar features in some way.

Neural-Gas (NGAS) is a VQ technique with soft competition between the units. In each training step, the squared Euclidean distances between a randomly selected input vector x_i from the training set and all *code-vectors* m_k are computed; the vector of these distances, expressed in (1) is d. Each centre k is assigned a rank $r_k(d) = 0, \ldots, N\text{-}1$, where a rank of 0 indicates the closest distant centre to x. The learning rule is expressed as it is indicated in (2).

$$d_{ik} = \|x_i - m_k\| = (x_i - m_k)^T * (x_i - m_k) \quad (1)$$

$$m_k = m_k + \varepsilon * h_\rho[r_k(d)] * (x - m_k) \quad (2)$$

$$h_\rho(r) = e^{(-r/\rho)} \quad (3)$$

A monotonically decreasing function of the ranking that adapts all the centers, with a factor exponentially decreasing with their rank is represented in (3). The width of this influence is determined by the neighborhood range ρ. The learning rule is also affected by a global learning rate ε. The values of ρ and ε decrease exponentially from an initial positive value ($\rho(0)$, $\varepsilon(0)$) to a smaller final positive value ($\rho(T)$, $\varepsilon(T)$) according to expressions (4) and (5) respectively, where t is the time step and T the total number of training steps, forcing more local changes with time.

$$\rho(t) = \rho(0) * [\rho(T)/\rho(0)]^{(t/T)} \quad (4)$$

$$\varepsilon(t) = \varepsilon(0) * [\varepsilon(T)/\varepsilon(0)]^{(t/T)} \quad (5)$$

2.2 The Collaborative Process

The collaborative process used for this research (Paletta et al, 08), (Paletta et al, 09-1), (Paletta et al, 09-2) is based on the concept of awareness of interaction. It has a CDE (E) containing a set of n nodes N_i ($1 \le i \le n$) and r items or resources R_j ($1 \le j \le r$). These resources can be shared as a collaborative mechanism among different nodes. It has:

1) N_i.*Focus*(R_j): It can be interpreted as the subset of the space (environment/ medium) on which the agent in N_i has focused his attention aiming for collaboration with, according to the resource R_j.

2) N_i.*NimbusState*(R_j): Indicates the current grade of collaboration that N_i can give over R_j. It could have three possible values: *Null*, *Medium* or *Maximum*. If the current grade of collaboration N_i that is given about R_j is not high, and this node could collaborate more over this resource, then N_i.*NimbusState*(R_j) will get the *Maximum* value. N_i.*NimbusState*(R_j) would be *Null* if there is not more collaboration possible with N_i related with R_j.

3) N_i.*NimbusSpace*(R_j): It Represents the subset of the space where N_i aims to establish the collaboration about R_j.

4) R_j.*AwareInt*(N_a, N_b): This concept quantifies the degree of collaboration over R_j between a pair of nodes N_a and N_b. It is manipulated via *Focus* and *Nimbus*, requiring a negotiation process. Following the awareness classification introduced by Greenhalgh (Greenhalgh, 97), values of this concept could be *Full*, *Peripheral* or *Null*.

5) N_i.*TaskResolution*(R_1,…,R_p): N_i requires collaboration with all R_j ($1 \le j \le p$).

6) N_i.*CollaborativeScore*(R_j): Determines the score to collaborate R_j in N_i. It is represented with a

value within [0, 1]. The closer the value is to 0 the hardest it will be for N_i to collaborate with the necessity of R_j.

Any node N_a in the CDE is represented by an agent that has the corresponding information about *E* (*Focus* and *Nimbus* for R_j). The collaborative process in the CDE follows these steps:

1) N_b must solve a task by means of a collaborative task-solving process making use of the resources $R_1,\ldots,R_p$, so that, it generates a N_b.*TaskResolution*($R_1,\ldots,R_p$).

2) N_b looks for the CDE current conditions to calculate the values associated to the key concepts of the model (*Focus*/*Nimbus* related to the other nodes), given by N_i.*Focus*(R_j) and N_i.*Nimbus*(R_j) $\forall i$, $1 \leq i \leq n$ and $\forall j$, $1 \leq j \leq r$.

3) Nodes in CDE respond to request for information made by N_b. This is done through the exchange of messages between agents.

4) As a final result of the previous information exchange the model will calculate the current awareness levels given by R_j.*AwareInt*(N_i, N_b).

5) N_b gets the collaboration score N_b.*CollaborativeScore*(R_j).

6) For each resource R_j ($1 \leq j \leq p$) included in N_b.*TaskResolution*($R_1,\ldots,R_p$), N_b selects the node N_a whose N_a.*CollaborativeScore*(R_j) is the most suitable to start the collaborative process (greatest score). Then, N_a will be the node in which N_b should collaborate on resource R_j.

7) Once N_a receives a request for cooperation, it updates its *Nimbus* (given by N_a.*NimbusState*(R_j) and N_a.*NimbusSpace*(R_j)).

8) Once N_a has finished collaborating with N_b it must update its *Nimbus*.

However, when conditions on the CDE are not appropriated enough to establish a collaboration process (N_i.*NimbusState*(R_j) = *Null* for most of the N_i, R_j) the conditions for collaboration are saturated. Therefore, if the node N_b initiates a collaborative process and find no more options to collaborate with and related to any R_j, then N_b could start a negotiation process that allows it to have new candidates to collaborate with and related to this specific R_j. Next section presents the details of this negotiation mechanism.

3 THE MAS-BASED NEGOTIATION MECHANISM

The negotiation mechanism proposed in this paper consists of three elements: 1) a heuristic algorithm for deciding the most suitable node to initiate negotiation based on current conditions; 2) a protocol for exchanging messages between agents; 3) a heuristic method to accept/decline a need for collaboration during a negotiation.

3.1 Deciding the Node to Negotiate

For deciding the most suitable node to negotiate with, the idea is to define a non-supervised learning strategy aiming to correlate the current information of the nodes in the distributive environment based on clusters. Most suitable node means a candidate that accepts the requirements necessary to collaborate with it.

To achieve the previous goal a NGAS-based algorithm is used. Therefore, the decision consists on identifying the node that is closest to the hyper-plane defined by the space given by the current environment conditions. In other words, it is necessary to determine the winning unit as a result of testing the NGAS with the environment.

Input vector is defined as follows (being N_b the node who requires collaboration on a set of resources and therefore who sends the N_b.*TaskResolution*($R_1,\ldots,R_p$), for each $N_a \neq N_b$):

1) The N_a.*NimbusState*(R_j) that will be represented by a value *Nst* within the interval [0,1] being *Nst* = 1 the value associated to N_a.*Nimbus State*(R_j) = *Maximum*, *Nst* = 0.5 the value associated to N_a.*NimbusState*(R_j) = *Medium*, and *Nst* = 0 the value associated to N_a.*NimbusState*(R_j) = *Null*.

2) The R_j.*AwareInt*(N_a, N_b) that will be represented by a value *AwI* within the interval [0,1] being *AwI* = 1 the value associated to R_j.*Aware Int*(N_a, N_b) = *Full*, *AwI* = 0.5 the value associated to R_j.*AwareInt*(N_a, N_b) = *Peripheral*, and *AwI* = 0 the value associated to R_j.*AwareInt*(N_a, N_b) = *Null*.

Therefore, the *code-vectors* for this problem have 2*n* elements, being *n* the number of nodes in the CDE. If $N_a = N_b$ then *Nst* = *AwI* = 0.

3.2 The Negotiation Protocol

Agents in the CDE exchange the following three messages (see Fig. 1 for this protocol):

1) *REQUEST*: Once N_b has indentified a node N_a to negotiate with, N_b uses this message to communicate its need to N_a so that N_a will accept to collaborate with N_b in relation to the resource R_j.

2) *CONFIRM*: In response to a *REQUEST* message, N_a uses this message to inform N_b that it has accepted the request for collaboration.

3) *DISCONFIRM*: In response to a *REQUEST*, N_a informs N_b that it has not accepted the request for collaboration.

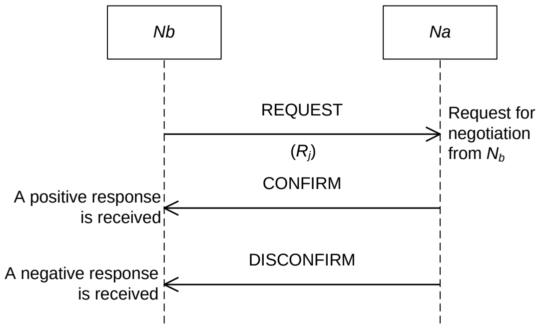

Figure 1: The inter-agent negotiation protocol.

Note that the ultimate goal of negotiation is to make a node accept a proposal to change its current condition provided by its *Nimbus*. On the other hand, in case of a negative response, N_b can decide between looking for another candidate to negotiate with and declining to seek collaboration in relation to the particular resource R_j.

3.3 Accept/Decline Collaboration

As with the decision of the most suitable node to negotiate with, this is also an ANN-based strategy. In this case there are r supervised ANN, one for each resource R_j defined in the environment. All ANNs are defined in the same way. There are three inputs and one output. The output $s \in [0, 1]$ represents the decision i.e. it is accepted if $s \geq 0.5$, and declined otherwise. Inputs are as follows:

1) A value $PhyAsp(R_j) \in [0, 1]$ that indicates the level of physical availability of the resource R_j; $PhyAsp(R_j) = 1$ means that the resource is completely available, $PhyAsp(R_j) = 0$ means that the resource is fully saturated.

2) A value equal to 1 if $N_b \in N_a.Focus(R_j)$, being N_b the node that is requiring for the decision, and N_a the node that should make the decision. If $N_b \notin N_a.Focus(R_j)$ then entry is 0.

3) A value equal to Nco_{NR} / $TNco(R, N)$, being Nco_{NR} the number of times N (node that is requiring for the decision) has collaborated with the current node (node that should make the decision) related to R. Therefore, Nco is a nxr matrix that should be updated by each node in the environment. The idea is to reward those nodes N_b that collaborated in the past with N_a and are just now requiring collaboration of N_a. $TNco(R, N)$ is calculated by using (6).

$$TNco(R,N) = \begin{cases} \sum_{j=1}^{r} Nco_{Nj}, & \sum_{j=1}^{r} Nco_{Nj} \neq 0 \\ \text{random}(Nco_{NR}, 1), & \text{otherwise} \end{cases} \qquad (6)$$

The ANNs used in this strategy are Multi-Layer Perceptrons (MLPs) based models. There is only one hidden layer with two units.

4 EVALUATION

A MAS used for CDEs has been created to evaluate the negotiation mechanism presented in this paper. Agents called IA-Awareness were defined by using the architecture SOFIA (SOA-based Framework for Intelligent Agents) (Paletta et al, 09-2), (Paletta et al, 09-3). This MAS-based platform has been implemented in JADE (Bellifemine et al, 99).

The evaluation of the mechanism was conducted in a TCP/IP-based LAN (Local Area Network) which assumes that each node (PC) can directly communicate with any other node. The experimentation was conducted by simulating different scenarios aiming to rate the capability of the method used for managing the growth of the nodes in the different environment conditions. The scenarios were defined by changing the quantity of nodes/PCs n (agents) as well as the number of resources r according to $n \in \{4, 8\}$ and $r \in \{2, 6, 10\}$. Therefore 6 different scenarios were simulated: 1) $n = 4$, $r = 2$; 2) $n = 4$, $r = 6$; 3) $n = 4$, $r = 10$; 4) $n = 8$, $r = 2$; 5) $n = 8$, $r = 6$; and 6) $n = 8$, $r = 10$. Moreover:

1) The initial condition of the CDE for each scenario ($N_i.Focus(R_j)$, $N_i.NimbusState(R_j)$ and $N_i.Nimbus\ Space(R_j)$; $1 \leq i \leq n$; $1 \leq j \leq r$) was randomly defined by considering the following: one node belongs to the *Focus* of another node with a probability of 0.75 and to the *Nimbus* with a probability of 0.85.

2) All N_b nodes execute an automatic process that generates $N_b.TaskResolution(R_1,\ldots,R_p)$ by randomly selecting the involved resources from the 50% of the total resources in the scenario.

3) $PhyAsp(R_j)$, $\forall j$ $1 \leq j \leq r$ were randomly initialized.

4) The parameters used for configuring the NGAS-based ANNs are the following: $\varepsilon(0) = 1.58$; $\varepsilon(T) = 0.02$; $\rho(0) = 5.59$; $\rho(T) = 0.07$.

Aiming to measure the effectiveness (θ) and efficiency (ξ) of the negotiation mechanism, expressions, (7) and (8) were defined respectively (note that both measures (θ, ξ) are positive values in

[0, 1] where 1 is the maximum effectiveness and efficiency). Where:

- *PSN* is the percentage of successful negotiations made in saturated conditions based on the number of negotiations that receive a positive response in relation to the total attempts.

- *MDN* is the mean duration in seconds of the negotiation process under saturated conditions. The process starts at the moment the node requires the cooperation until it receives an answer, whether affirmative or negative.

- *ATC* is the average time of collaboration in seconds calculated since $TaskResolution(R_1,\ldots,R_p)$ starts until it ends.

$$\theta = PSN / 100 \qquad (7)$$

$$\xi = 1 - MDN / ATC \qquad (8)$$

Table 1: Measures obtained from simulation of each scenario.

Measure	$n = 4$ $r = 2$	$n = 4$ $r = 6$	$n = 4$ $r = 10$	$n = 8$ $r = 2$	$n = 8$ $r = 6$	$n = 8$ $r = 10$
PSN	100,00	68,13	64,29	93,75	78,13	100,00
MDN	0,00	2,14	1,23	0,13	2,16	0,44
ATC	3,40	3,46	3,34	7,47	14,87	19,28
θ	1,00	0,68	0,64	0,94	0,78	1,00
ξ	1,00	0,38	0,64	0,98	0,85	0,98

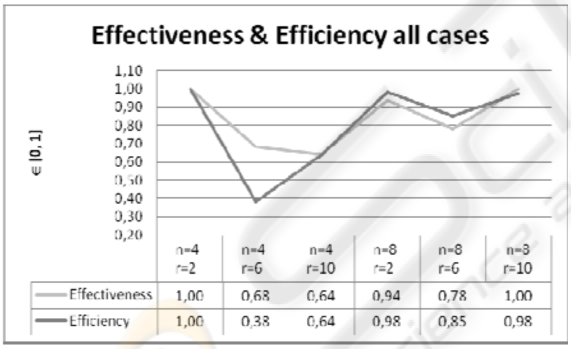

Figure 2: Results obtained from simulations.

Table 1 shows the measures obtained after a simulation of 120 minutes for each scenario, and Figure 2 shows the effectiveness and efficiency related with these measures. According to these results it is possible to make the following observations and/or conclusions:

1) The average effectiveness is 0,84 and the average efficiency is 0,81.

2) Both effectiveness and efficiency have a similar trend of behavior.

3) Nor the variation in the number of nodes or the variations in the number of resources have a particular tendency to improve or worsen the effectiveness and efficiency.

It is important to stress that, due to the fact that it is a learning-based mechanism from past situations, it is assumed that as there is much more to learn, the metrics associated with it must be improved.

5 RELATED WORK

Regarding the context of awareness and recognizing the current context of a user or device, authors in (Mayrhofer et al, 07) present an approach based on general and heuristic extensions to the growing NGAS algorithm classifier which allow its direct application for context recognition.

The use of ANN technology for negotiation algorithms can be found in (Oprea, 02), (Roussaki et al, 07), (Zeng et al, 05), (Sakas et al, 07). Author in (Oprea, 02) presents an adaptive negotiation model that uses a feed-forward artificial neural network as a learning capability to model the other agent negotiation strategy. In (Roussaki et al, 07), authors proposed a MLP-based learning strategy that is used mainly to detect at an early stage the cases where agreements are not achievable, supporting the decision of the agents to withdraw or not from the specific negotiation thread.

In the same order of ideas, authors in (Zeng et al, 05) propose an agent-based learning method in automated negotiation based on ANN aiming to implement interactions between agents and guarantee the profits of the participants for reciprocity. Finally, authors in (Sakas et al, 07) overcome the difficulty of using fuzzy logic and fuzzy neural networks by applying an adaptive neural topology to model the negotiation process.

Although the use of ANN for negotiation mechanisms can be found in several previous works, as far as we know, there is no similar approach related with the subject of this paper: a non-supervised based model for learning cooperation on CDE by using the awareness concept.

6 CONCLUSIONS AND FUTURE WORK

This paper presents a new negotiation mechanism used for a MAS-based system that is a part of Collaborative Distributed Environments (CDEs). The method proposed is endowed with two heuristic algorithms and an exchange message protocol between agents. The heuristic algorithms are used primarily for deciding the most suitable node to collaborate with, and secondly, for the agent that receives a request for negotiation to decide whether or not to choose if it wants/can to collaborate.

Results show that the mechanism has an average effectiveness of 0,84 and an average efficiency of 0,81. Therefore, this mechanism ensures an agreement in negotiation in a short period of time.

Although this method has not yet been tested in a real CDE, it has been designed to be suitable for real environments. In fact the validation carried out to presently demonstrate that this method could be extended to real scenarios in CDE with no problems.

We are currently working on testing this method in real CDE as well as using this strategy in grid and cloud computing environments.

REFERENCES

Andre, D., Koza, J., 1996. A parallel implementation of genetic programming that achieves super-linear performance. In *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 1163-1174.

Arenas, M.I., Collet, P., Eiben, A.E., Jelasity, M., Merelo, J.J., Paechter, B., Preuss, M., Schoenauer M., 2002. A Framework for Distributed Evolutionary Algorithms. In *Proc. 7th International Conference on Parallel Problem Solving from Nature (PPSN VII)*. LNCS 2439, pp. 665-675.

Bellifemine, F., Poggi, A. Rimassa, G., 1999. JADE – A FIPA-compliant agent framework. *Telecom Italia internal technical report*. In *Proc. International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAM'99)*, pp. 97-108.

Blanchard, E., Frasson, C., 2002. Designing a Multi-agent Architecture based on Clustering for Collaborative Learning Sessions. In *Proc. International Conference in Intelligent Tutoring Systems*, LNCS. Springer.

Greenhalgh C., 1997. Large Scale Collaborative Virtual Environments. Doctoral Thesis, *University of Nottingham*.

Ferber, J., 1995. Les systems multi-agents, *Vers une intelligence collective*. Ed. InterEditions, pp. 1-66.

Kohonen, T., Mäkisara, K., Saramäki, T., 1984. Phonotopic maps – insightful representation of phonological features for speech recognition. In *Proc. 7th International Conference on Pattern Recognition*, pp. 182-185.

Lin, F., Lin, Y., 2006. Integrating multi-agent negotiation to resolve constraints in fulfilling supply chain orders *Electronic Commerce Research and Applications*, Vol. 5, No. 4, pp. 313-322.

Makhoul, J., Roucos, S., Gish, H., 1985. Vector quantization in speech coding. In *Proc. IEEE 73*, IEEE Computer Society, pp. 1551-1588.

Martinetz, T.M., Schulten, K.J., 1991. A neural-gas network learns topologies. In T. Kohonen et al (eds)., *Artificial Neural Networks*, pp. 397-402.

Mayrhofer, R., Radi, H., 2007. Extending the Growing Neural Gas Classifier for Context Recognition. Computer Aided Systems Theory – *EUROCAST 2007*, pp. 920-927.

Nasrabadi, N.M., Feng, Y., 1988. Vector quantization of images based upon the Kohonen self-organizing feature maps. In *IEEE International Conference on Neural Networks*, pp. 1101-1108.

Nasrabadi, N.M., King, R.A., 1988. Image coding using vector quantization: A review. *IEEE Trans Comm* 36(8), IEEE Computer Society, pp. 957-971.

Naylor, J., Li, K.P., 1988. Analysis of a Neural Network Algorithm for vector quantization of speech parameters. In *Proc. First Annual INNS Meeting, NY*, Pergamon Press, pp. 310-315.

Oprea, M., 2002. An Adaptive Negotiation Model for Agent-Based Electronic Commerce. *Studies in Informatics and Control*, Vol. 11 i3, pp. 271-279.

Paletta, M., Herrero, P., 2008. Learning Cooperation in Collaborative Grid Environments to Improve Cover Load Balancing Delivery. In *Proc. IEEE/WIC/ACM Joint Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 399-402.

Paletta, M., Herrero P., 2009. Awareness-based Learning Model to Improve Cooperation in Collaborative Distributed Environments. In *Proc. 3rd International KES Symposium on Agents and Multi-agents Systems – Technologies and Applications (KES-AMSTA 2009)*, LNAI 5559, Springer, pp. 793-802.

Paletta, M., Herrero P., 2009. Foreseeing Cooperation Behaviors in Collaborative Grid Environments. In *Proc. 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'09)*, Vol. 50/2009, Springer, pp 120-129.

Paletta M., Herrero P., 2009. Towards Fraud Detection Support using Grid Technology, *Special Issue at Multiagent and Grid Systems* - An International Journal, Vol. 5, No. 3, IOS Press, pp. 311-324.

Roussaki, I., Papaioannou, I. Anangostou, M., 2007. Building Automated Negotiation Strategies Enhanced by MLP and GR Neural Networks for Opponent Agent Behaviour Prognosis. *Computational and Ambient Intelligence*, V. 4507, Springer, pp. 152-161.

Sakas, D.P., Vlachos, D.S., Simos, T.E., 2007. Adaptive Neural Networks for Automatic Negotiation. In *Proc. of the International Conference on Computational Methods in Science and Engineering (ICCMSE 2007)*, Vol. 2, Parts A and B, AIP Conference Proceedings, pp. 1355-1358.

Wooldridge., M.J., 2002. An Introduction to Multiagent Systems, *John Wiley and Sons*.

Zeng, Z.M., Meng, B., Zeng, Y.Y., 2005. An Adaptive Learning Method in Automated Negotiation Based on Artificial Neural Network. In *Proc. of 2005 International Conference on Machine Learning and Cybernetics*, Vol. 1, Issue 18-21, pp. 383-387.