A NEW APPROACH IN LEARNING FOR INTELLIGENT MULTI AGENT SYSTEMS

Ahmed M. Elmahalawy
Department of Cybernetics,
Czech Technical University,
Karlovo námestí 13, 121 35 Prague 2, Czech Republic.
E-mail: elmahal@labe.felk.cvut.cz

KEYWORDS
Intelligent Agent Technology, Intelligent Multi Agent System, Machine Learning Algorithm, Repertory Grid Algorithm.

ABSTRACT
The agent technology has recently become one of the most vibrant and fastest growing areas in information technology. This technology is developed to use more than one agent in the system; this is called Multi Agent systems (MAS). The system that depends on this technology should have been studied extensively. One of the most important characteristic of this is its ability to learn and adapt itself, where it has been done using one of the machine learning algorithms. Repertory Grid (RG) which has become a widely used and accepted technique for knowledge elicitation and has been implemented as a major component for many computer-based knowledge acquisition systems. In this paper, RG has been developed to be one of the machine learning algorithms and then used in MAS.

INTRODUCTION

The Artificial Intelligence (AI) is one of the most important branches of computer science, because it tries to simulate the human reasoning. It includes many branches such as: Expert System, Neural Network, Robotics etc. The new and rapidly growing branch is the Intelligent Agent System (IAS). Agent technology has received a great deal of attention over the last few years and, as a result, the industry is beginning to get interested in using this technology to develop its own products [Ali 1996, Chorafas 1998, Nwana and Ndumu 1999].

Many previous works in this field are related to analyze a popular machine learning algorithms as ID3, C4.5 and UNIMEM. Our work is to improve RG to be a machine learning algorithm and then make an analysis for it when it learns MAS. RG technique plays a central role in the elicitation methodology of many well-reported knowledge acquisition tools or workbenches. However, the dependability of these systems is low where the technique breaks down or proves inadequate due to limited expressive power and other problems [Batty and Kamel 1995, Shaw and Gaines 1990].

This paper begins by introducing the Multi Agent Systems (MAS) architecture and appropriate functional for each component of this architecture. In section 3, the Repertory Grid techniques has been analyzed and representing its development to be machine learning algorithm. The basic Repertory Grid algorithm is presented in section 4. In section 5 , the performance of the system is evaluated by using Repertory Grid algorithm. A group of figure of practical results is shown in section 6. Finally, a conclusion is described in section 7.

THE MULTI AGENT ENVIRONMENT

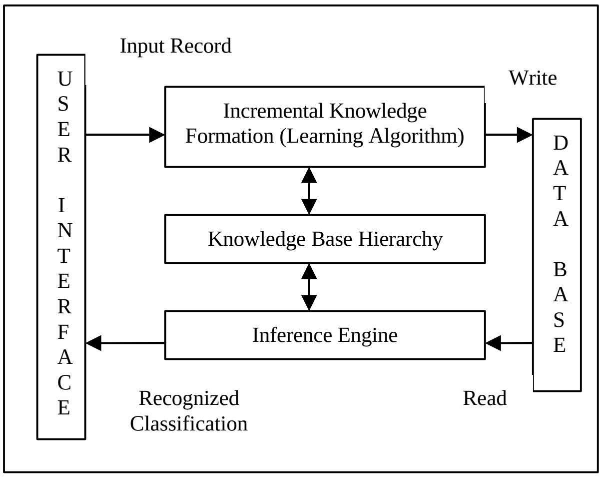
It has been noted that MAS consists of a group of agents that can cooperate to solve the problem. In building MAS software, there are two important things to be considered, the agent model and the method of cooperation among agents. In the following, the discussion about the used architecture for single agent and multi agent systems is presented. For the method of cooperation among agents, the MAS uses the communication protocols among agents called contract net protocol [Pradromidis 1998, Smith and Davis 1988, Weiss 1999]

The layered architecture is used in this work to build the agent. The layered architecture is structured into a number of layers each of which typically represents a particular function. An example of this architecture is the three layer architecture. These layers are concept formation layer, knowledge sources layer, and inference engine layer [Richards 2004, Saravanan and Vivekanandan 2004]. Based on this architecture, an agent is built where it consists of three main subsystems that are incremental knowledge formation process, the knowledge base hierarchy and the inference engine. Figure 1 represents the basic architecture of the intelligent agent and the different interactions within its internal concepts and the external entities as well.

It is clear that many architectures exits for MAS. Those architectures are fully dependent on the kind of agents in the system. For most systems, homogenous agents are used but there is an increasing interest in heterogeneous agents. We will present MAS architecture. The single agent is the basic unit in MAS. We built our basic architecture of the intelligent agent depending on three

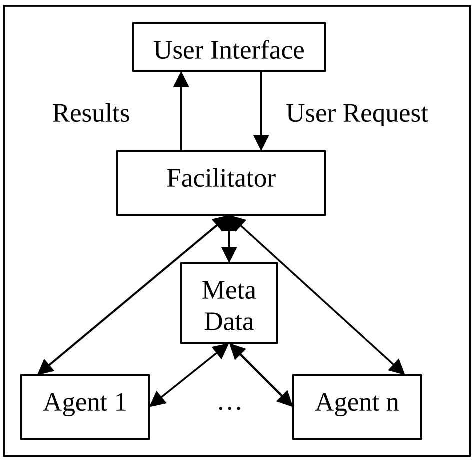
Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

main subsystems, incremental knowledge formation process, the knowledge base hierarchy and the inference engine [Richards 2004].



Figures 1: The Components of the Intelligent Agent

The used MAS architecture is considered as a collection of homogeneous agents that are globally controlled by the facilitator depending on the principle of contract net protocol, as shown in Figure 2.



Figures 2: MAS Architecture

The user interface presents data (results) to the user and collects user request. The facilitator coordinates the agents, presents information to the user interface, and provides response to the agents from the user. That means that the facilitator is responsible for distribution of the data to each agent according to each agent domain and collects the results from agents. We can say that the facilitator is the system controller. Meta data simply means "data about data", where the basic metadata is the database schema; that is, and the physical layout of the data in tables. The agent is responsible for results according to the learning from the pervious data [Zhang et al. 2005].

OVERVIEW OF REPORTY GRID ALGORITHM

RG is a well-known knowledge acquisition and representation technique based on the work of Kelly on personal construct theory. It has been applied to a wide variety of domains, usually aimed at various kinds of heuristic classification or expert system formation. Their general applicability makes them very attractive in knowledge acquisition (KA). It is therefore a natural step to seek ways to increase their power in general knowledge-based reasoning [Delugach and Lampkin 2000].

RG consists of a rectangular matrix of ratings of things called elements (usually placed in the columns) each rated on adjectival phrases or simple adjectives known as constructs. There are a number of ways of using such matrices to throw light on the respondent's construing. Analyses are facilitated by computer programs and it is possible to carry out both qualitative/idiographic and homothetic research using grids as the basic tool. In other word, RG is a two way classification of data in which events are associated with abstractions in such a way as to reveal the relationship between persons' observations of the world (also called elements) and their construct or classification of those experiences. The elements are the things that are used, first, to define the area of the topic, and second, to determine the universe of discourse. The constructs are the terms in which the elements are similar or different from each other. Constructs have two poles, each of which has a meaning concerning its opposite [Delugach and Lampkin 2000].

The aim of this technique is to eliminate the knowledge engineer figure, and with him/her all the communication problems, and encourage the expert, dealing with a knowledge support system, to communicate his/her knowledge directly to the system.

RG is well-established as a general and powerful knowledge elicitation and acquisition technique to support classification. Its strengths are [Delugach and Lampkin 2000]:

1- A solid foundation in human psychological theory.
2- Demonstrated utility in eliciting and acquiring knowledge from people.
3- A general applicability in diverse domains.
4- Freedom from specific paradigms or observer bias.
5- An ability to acquire knowledge of a non-discrete nature (i.e., poles possessing a continuous set of values).

It is possible to assess the similarity of known elements. There are three different distance measures provided. In any of the given measures, the lower the distance between two elements, the more similar the elements are deemed to be. The measures are as the following laws [Black 2006]:

Euclidean Distance

The number of attributes defines the number of dimensions of the space. The distance between two elements in a single dimension is the distance between the values of an attribute. To find the distance in n dimensions, use the Euclidean distance:

$$S = \sqrt{\sum_{i=1}^{n} (Xi - Yi)(Xi - Yi)} \tag{1}$$

Manhattan Distance

This measure simply sums the distances in each dimension, so the distance measure in n dimensions is the following

$$S = \sum_{i=1}^{n} |(Xi - Yi)| \tag{2}$$

Hamming Distance

This measure gives a value of 0 or 1 to distance between two elements in any one dimension. The distance is 0 if the assigned values are the same and 1 otherwise. The hamming distance over n dimensions is therefore the number of attributes which have different values:

$$S = \sum_{i=1}^{n} F(X_i, Y_i) \tag{3}$$

$$\text{where } F(X_i, Y_i) = 0 \text{ if } X_i = Y_i \text{ else } F(X_i, Y_i) = 1$$

This similarity distance is very important for that algorithm, where it depends on it.

THE BASIC ALGORITHM STEPS

As we have seen that RG is a knowledge elicitation and acquisition technique, we enhance RG technique to allow it t use as a machine learning algorithm, which is based on the following two concepts:

1- Building a hierarchy as in UNIMEM algorithm, such as adding and deleting a node according to a particular parameter(s).

2- Deducing the similarity between the nodes in the grid. This can be measured by one of the similarity distance (S) such as Euclidean, Manhattan and Hamming distance.

The basic steps for the new RG algorithm are represented as follows:

1- Enter the T training set which is a full grid

2- Adjust the algorithm parameters S, $B4$.

3- In each column, assign the weight of each value according to the different values in this column, i.e. if we have three different values in the column so we have a scale of 3 and each value take a number from this scale.

4- Consider the first node is one pole of the grid and each input node is compared to it to see if this node will be included with this pole or another pole.

5- Calculate the similarity distance between the two nodes, using one of the similarity distances.

6- If the similarity distance between two nodes is more than S then delete the feature with the largest weight in new node.

7- If the number of the features in the new node is less than $B4$ then delete this node and assign it to another grid pole.

8- Repeat the evaluation of similarity distance until this evaluation is less than or equal to S.

9- For each new node repeat the steps 6 to 8 to assign it to one of the two poles.

The pseudo-code of RG algorithm is presented in Figure 3.

```
Repertory Grid Algorithm [T  : set of records (full grid), S:
similarity distance, B4: the minimum number of features to
retain the node]
Begin
 Adjust the algorithm parameters S, B4
 Calculate the scale's value for each features in the node of the
grid initialize the two poles of the grids n=1, m=2
 Do
     calculate the similarity distance between the two nodes
     If similarity distance is more than S Then
        delete the feature with the largest scale in the new node,
        repeat the evaluation of the similarity distance and the
      comparison until similarity distance is less than or equal S
     End If
     If the number of the feature in the node is less than B4
    Then delete the node and exchange with the last full node
           in the grid
     Else
    If you reach to the first node then use the next node
                                              in the grid
           End If
     End  If-Else
End Do
End RG.
```

Figures 3: The RG Algorithm

The following example is a simpler example from the stock market involving only the discrete ranges of a profit as a goal attribute, with values {up, down}. Table 1 shows the full grid of this data with the weight of each feature in the node.

The algorithm is applied on the previous example by entering one record each time. The algorithm uses Hamming distance to calculate the similarity distance, and the following parameters S=1 and $B4$ = 2. The following tables (2 to 5) indicate the way to reach to final table when all the records of training set are entered.

Table 1: The Full Grid

Profit	Type		Competition		Age	
	W.	V.	W.	V.	W.	V.
down	1	swr	1	Yes	1	old
down	1	swr	2	No	1	old
down	2	hwr	2	No	1	old
down	1	swr	1	Yes	2	mid
down	2	hwr	1	Yes	2	mid
up	2	hwr	2	No	2	mid
up	1	swr	2	No	2	mid
up	1	swr	1	Yes	3	new
up	2	hwr	2	No	3	new
up	1	swr	2	No	3	new

Table 2 presents the data in the table after the record No.3 is entered. Table 3 presents the data in the table after the record No.5 is entered. Table 4 presents the data in the table after the record No.7 is entered. Table 5 presents the data in the table after the record No.10 is entered.

Table 2: The Data Table after 3 Records Are Entered

Profit	Type		Competition		Age	
	W.	V.	W.	V.	W.	V.
down	1	swr	1	Yes	1	old
down	1	swr	2	No	1	old
down	2	hwr	-	-	1	old

Table 3: The Data Table after 5 Records Are Entered

Profit	Type		Competition		Age	
	W.	V.	W.	V.	W.	V.
down	1	swr	1	Yes	1	old
down	1	swr	2	No	1	old
down	2	hwr	-	-	1	old
down	1	swr	1	Yes	2	mid
down	2	hwr	1	Yes	2	-

Table 4: The Data Table after 7 Records Are Entered

Profit	Type		Competition		Age	
	W.	V.	W.	V.	W.	V.
down	1	swr	1	Yes	1	Old
down	1	swr	2	No	1	Old
down	2	hwr	-	-	1	Old
down	1	swr	1	Yes	2	Mid
down	2	hwr	1	Yes	2	-
up	2	hwr	-	-	2	Mid

Table 5: The Data Table after 10 Records Are Entered

Profit	Type		Competition		Age	
	W.	V.	W.	V.	W.	V.
down	1	swr	1	Yes	1	old
down	1	swr	2	No	1	old
down	2	hwr	-	-	1	old
down	1	swr	1	Yes	2	mid
down	2	hwr	1	Yes	-	-
up	2	hwr	-	-	2	mid
up	1	swr	1	Yes	3	new
up	1	swr	2	No	-	-

The previous table contains the data obtained from applying the RG algorithm on the previous example as it has seen, that data is reduced and to be less than the data in Table1

ALGORITHM ANALYSIS

How can architecture be evaluated? Evaluation of architecture can take different forms, depending on one's interests. For instance, someone with a practical objective would be primarily interested in observable performance. This could include multiple dimensions of evaluation, involving input-output mappings, speed, running costs, generality, precision, accuracy, adaptability [Franklin 2002]. The following steps are used to evaluate the system performance. The system is learned with a sample of records called training records in which its result is known and measure the performance of the system using the viability (the view ability of the system to learn from these records), that is done by using another sample of records called testing records whose result is also known. This measurement is defined as shown.

Viability is the positive scale, initially takes the value 0 that represents the number of testing records that the MAS can give a correct or wrong result for it when we enter the testing record No. X, e.g., when we enter record No. 1 if the system gives the correct result for it so we increase the viability by one, else there is no change, when we enter record No. 2, if the system gives the correct result for it, so we increase the viability by one, else we decrease the viability by one if it has a value not equal to zero. So, the value of the viability at test record No. 2 represents the number of correct and wrong results for this record and the pervious records [Nwana and Ndumu 1999].

We can represent this scale as the following:

Initially: Viability = 0

$$\textbf{Viability} = \begin{cases} \textbf{Viability + 1} & \text{If the system result = actual result} \\ \textbf{0} & \text{If the system result} \neq \text{actual result and Viability =} \\ \textbf{Viability - 1} & \text{If the system result} \neq \text{actual result and Viability} \neq 0 \end{cases} \quad (4)$$

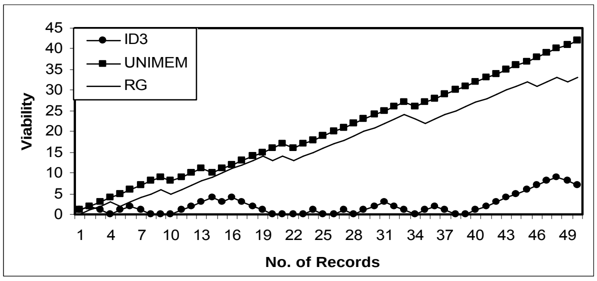
Note: there is no meaning for negative number of that scale.

THE PRACTICAL RESULTS OF REPERTORY GRID

Our practical results consists two parts, the first one: is showing the comparison between RG and a well known machine learning algorithm, ID3 and UNIMEM. We see the results in both Single and Multi Intelligent Agent Systems, Where the system have been applied to the training database collected for admission of fresh graduates to ITI (Information Technology Institute).

The second phase: analysis of the RG by changing the parameters that were mentioned in previous section. In each case, we change the algorithm parameters and drawing the viability curve. These parameters are changed as follows: S = 1, 2, 3 & $B4$= 2, 3, 4 and the similarity distance is one of the three cases (Euclidean, Hamming, and Manhattan distance). From this analysis, we need to indicate the most suitable parameters for the algorithm to give us the best result.

Figures 4 and 5 present the using the three different machine learning algorithms in both Single and Multi Intelligent Agent Systems respectively.



Figures 4: IAS using ID3, UNIMEM and RG Algorithms

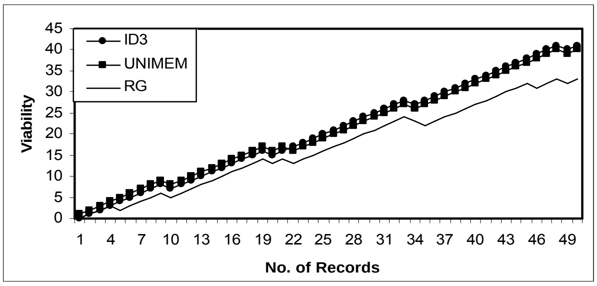
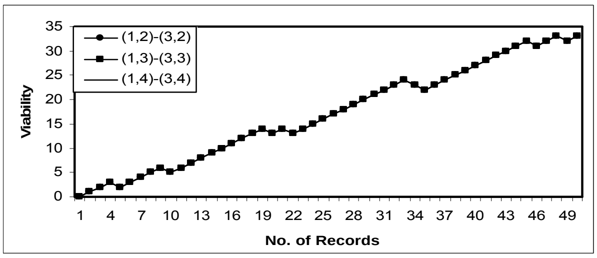


Figure 5: MAS (2 agents) using ID3, UNIMEM and RG Algorithms

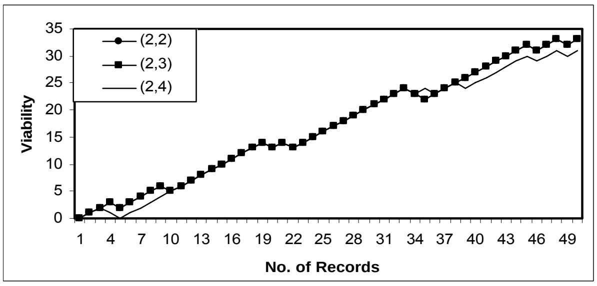
As we see from the last two figures, that RG gives a promising performance as the machine learning algorithms.

Figures 6 and 7 present the sample of the practical results of RG algorithm when changing its parameters and applying the problem of admission the graduated in Information Technology Institute (ITI).
Figure 6 shows that there is similarity in the system performance in case of different parameters. So, the changing of the parameters doesn't effect on the number of correct result.



Figures 6: RG Algorithm Using Euclidean and Manhattan Distance

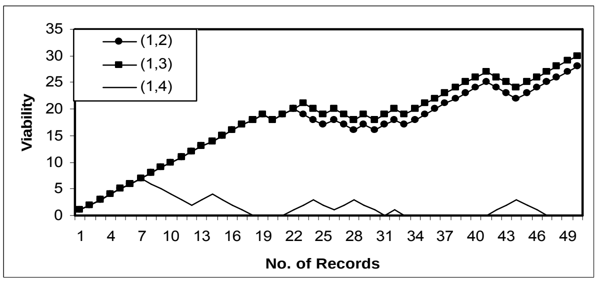
Figure 7 presents that the best system performance is occurred when using parameters (2, 2) and (2, 3) because the large number of correct results is obtained with these parameters values. When using parameters (2, 4), the number of correct results are near the previous case.



Figures 7: RG Algorithm Using Hamming Distance

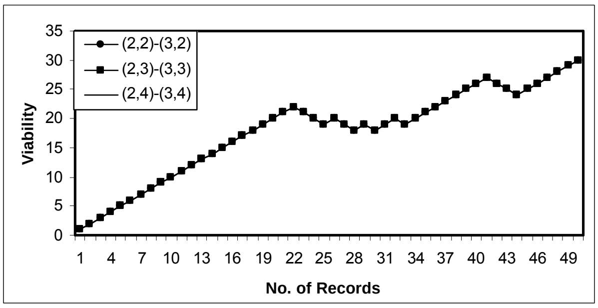
Figures 8 and 9 illustrate the sample of the practical results of RG algorithm when changing its parameters and applying the taking off and landing of airplane problem.

Figure 8 illustrates that the best system performance is occurred when using parameters (1, 3) and then (1, 2) because they give approximately large number of correct results. But using parameters (1, 4), the performance of the system is bad because the number of correct results are low.



Figures 8: RG Algorithm Using Euclidean Distance

Figures 9 illustrates that the system performance is the same for all parameters because the changing of the parameters doesn't effect on the number of correct result.



Figures 9: RG Algorithm Using Hamming and Manhattan Distance

CONCLUSION AND FUTURE WORK

At first, we see that the Repertory Grid gives a good performance as a well known machine learning algorithms, ID3 and UNIMEM. Then, according to the performance of Repertory Grid algorithm mentioned in the previous section. Table 6 summaries the results, where it gives the idea of the best parameters to give good results with this algorithm. These results have been discussed according to the percentage number of correct record (NOCR).

Table 6 shows that for most cases the best performance occurs when we use the RG parameters (S, B4) = (3, 4) and the Euclidean Distance as a similarity distance. RG can be considered as one of the machine learning algorithm according to the displacement measurements.

Table 6: Summary of Repertory Grid Performance

	Admission Problem	**Airplane Problem**
Euclidean Distance	The best performance occurs when the RG parameters (S,B4) are (3,3) and (3,4) where the percentage of NOCR = 84 %	The best performance occurs when the RG parameters (S,B4) are (3,2) and (3,3) and (3,4) where the percentage of NOCR = 86 %
Manhattan Distance	The performance is same for all RG parameters where the percentage of NOCR = 82 %	The best performance occurs when the RG parameters (S,B4) is (1,3) where the percentage of NOCR = 80 %
Hamming Distance	The best performance occurs when the RG. parameters (S,B4) are (3,4) where the percentage of NOCR = 84 %	The best performance occurs when the RG. parameters (S,B4) are (3,2) and (3,3) and (3,4) where the percentage of NOCR = 82 %

We as saw that RG Algorithm gives the promising results. Our future work is to study this algorithm in more details and try to put it in slandered form to be as one of the machine learning algorithms.

REFERENCES

Ali M. 1996. Intelligent-Agent Based Decision Support. A Thesis submitted in Partial Fulfillment for the degree of Master of Science, Cairo University, Faculty of Engineering, Egypt.

Batty D. and Kamel M.S. 1995. Automating Knowledge Acquisition: A Propositional Approach to Representing Expertise's an Alternative to Repertory Grid Technique. IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 1,53-67.

Black P.E. 2006. Dictionary of Algorithms and Data Structures. [WWW] Available from: http://www.nist.gov/dads/. [Accessed: 31 January 2007]

Chorafas D.N. 1998. Agent Technology Handbook. McGraw-Hill, New York USA.

Delugach H.S. and Lampkin B.E. 2000. Troika: Using Grids, Lattices and Graphs in Knowledge Acquisition. In Working with Conceptual Structures: Contributions to ICCS 2000 ,G. Stumme, Ed. Aachen, Germany: Shaker Verlag, 201-214.

Franklin S. 2002. Automating human information agents. In Intelligent agents and their applications, 27 – 58, Physical-Verlag GmbH Heidelberg, Germany, Germany

Nwana H.S. and Ndumu D.T.1999. A Perspective on Software Agents Research. In The Knowledge Engineering Review, Vol.14, No.2,1-18.

Pradromidis A.L. 1998. On The Management of Distributed Learning Agents. Ph.D. Thesis Proposal, Department of Computer Science, Columbia University, New York, USA.

Richards R.M. 2004. A visual, hierarchical approach to implementing rule-based algorithms in classification of discrete, homogenous objects. Decision Support Systems, Vol.38 , No.1,81 – 89.

Saravanan V. and Vivekanandan K. 2004. Design and implementation of automated data mining using intelligent agents in object oriented databases. In Intelligent information processing II, 221-226, Springer-Verlag London, UK

Shaw M.L.G. and Gaines B.R. 1990. Notes on Personal Construct Psychology and the Repertory Grid. http://www.cpsc.ucalgary.ca/~robertof/ courses/679/Notes.html.

Smith R.G. and Davis R. 1988. Frameworks For Cooperation in Distributed Problem Solving. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA .

Weiss G. 1999. Multi agent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press Cambridge, Massachusetts.

Zhang C. Zhang Z. and Cao L. 2005. Agents and Data Mining: Mutual Enhancement by Integration. In Lecture Notes in Computer Science, Vol.3505, 50-61. Springer Berlin / Heidelberg, Germany.

Ahmed M. Elmahalawy was born in Benha, Egypt and went to the Menoufia University, where he studies Computer Science and Engineering and obtained his degree in 1995. He worked as a demonstrator in Faculty of Electronic Engineering, Menoufia University. He had his MSc. in 2001and then worked as assistance lecture in Faculty of Electronic Engineering, Menoufia University. Now he is a PhD student in the Czech Technical University, department of Cybernetic.