Strategies for Exploiting Trust Models in Competitive Multiagent Systems

Víctor Muñoz, Javier Murillo, Beatriz López, and Dídac Busquets

Institut d'Informàtica i Aplicacions
Campus Montilivi, edifice P4, 17071 Girona
{vmunozs,jmurillo,blopez,busquets}@eia.udg.edu

Abstract. Multi-agent systems where agents compete against one another in a specific environment pose challenges in relation to the trust modeling of an agent aimed at ensuring the right decisions are taken. A lot of literature has focused on describing trust models, but less in developing strategies to use them optimally. In this paper we propose a decision-making strategy that uses the information provided by the trust model to take the best decisions to achieve the most benefits for the agent. This decision making tackles the exploration versus exploitation problem since the agent has to decide when to interact with the known agents and when to look for new ones. The experiments were performed using the ART Testbed, a simulator created with the goal of objectively evaluate different trust strategies. The agent competed in and won the Third International ART Testbed Competition held in Estoril (Portugal) in March 2008.

Key words: Competitive multi-agent systems, Trust, Reputation, ART Testbed

1 Introduction

There are, nowadays, various multi-agent environments in which agents act independently and compete to obtain the maximum benefits for themselves. These environments are usually composed of a large number of self-interested agents which interact with each other, offering and requesting services, in order to improve their performance. In order to decide which agents to interact with, agents generally use a built-in model of the other agents. This model gives the agent the information it needs to know which agents to *trust* in order to accomplish its objectives, as well as which agents to avoid. This is called the trust model of the agent [12].

The way an agent uses the trust model is similar to the way a human does. For instance, let us imagine a person, called *John*, who wants to buy a television. Last year John bought a television at the *Cheap & Good Store*, but shortly afterwards the television broke down. Consequently, John now has little trust in the *Cheap & Good Store*, and when it comes to buying another television he will prefer to buy it at another store. On the other hand, a relative of *John*, *Joe*, bought a television that worked well at the *Best TV Store* and he is therefore very pleased with the purchase and has a lot of trust in the store he bought it from. *Joe* mentions this to *John* who, as he trusts his relative, also has more trust in the *Best TV Store* (although *John* himself has not had any interaction with it). Finally, the *Best TV Store* can make publicity of itself, for example through advertisements.

As seen in the example, three different kinds of interactions can change the trust of an agent: (i) *direct trust* is based on personal experience, (ii) *indirect trust*, also called reputation, is based on another's experience of a third party, and (iii) *self trust* is based on an agent's advertising of itself. Although direct trust is in principle the most accurate, it is generally achieved at a higher cost (in terms of money, time, etc.) compared with the other kinds of trust [5]. Therefore, the use of indirect and self trust is indispensable in large environments in which it is not always possible (or it is too costly) to obtain direct trust. In such competitive environments, however, the use of indirect and self trust may be harmful, since some agents can act malevolently, given that a loss for a competitor may imply a benefit for itself. In fact, as a result of competitiveness it is quite likely that a considerable number of agents will not be honest, and try to deliberately produce losses in other agents in order to benefit from that behavior [16]. There is a distinction, though, between acting malevolently and giving bad quality service. To follow on from the previous example, a particular store may sell products of a higher quality than another, but this does not necessarily mean that the other store is acting malevolently. Conversely, a person may have a bad opinion of a particular store based on a bad past experience, and another person may have a good opinion of the same store based on a good past experience. If a third person asks them their opinions of the store, they will give opposing answers, but neither of them will be lying.

Therefore, the key requirement if an agent is to perform well is for it to have a trust model that takes into account all the above factors. However, a perfect trust model is difficult to obtain, and therefore the design of a strategy to take correct decisions based on an incomplete trust model (which may contain mistakes) is also an important factor if the agent is to obtain maximum benefits. In the decision-making process, the agent has to deal with the exploration versus exploitation problem [15], because when it wants to request a service it must decide whether to request it from one of the known agent providers (of services), or to explore some unknown ones and possibly discover better provider agents. If the agent is only focused on exploitation it might obtain good short-term results but bad ones in the long term, since exploration allows the agent to discover better provider agents than its normal providers, or to have substitute providers in case one of the usual providers fails (i.e. leaves the market or decreases in quality), a circumstance that is quite likely to happen when the agent is working in dynamic environments. Furthermore, exploration allows the agent to adapt to changes faster, given that if the quality of a provider diminishes for some reason, the agent will be able to change to another. Therefore, as an agent is usually limited in the number of interactions it can perform (due to time limits and interaction costs), it is necessary to decide which interactions to dedicate to exploration, and which to exploitation.

In this paper we assume that a trust model has been previously selected and we present a strategy to use it, dealing with the exploration versus exploitation problem. To test the system, the ART (Agent Reputation and Trust) Testbed simulator is used [4]. The aim of this testbed is to provide a standard problem scenario which can be used to compare different approaches to modeling and applying trust and strategies in multi-agent systems.

This paper is structured as follows. In the following section we survey existing trust models available in the literature. Next, the strategy for using it is designed. Later, we use the ART Testbed simulator to test the strategy. Finally, some conclusions are given.

2 Related Work

Trust has been defined in different ways for different domains. For example, one of the most frequently used is the definition of trust as "confidence in the honesty or goodness of an agent". In this case trust is measured by the behavior of an agent, based on a decision about whether it is acting honestly or maliciously. However, the definition in [5] is the most useful for our purposes: "Trust is the confidence or belief in the competence, utility or satisfaction expected from other agents concerning a particular context". Under this definition, the trust in an agent to provide a specific service can be understood as an estimator of the expected quality that will be obtained from that service. For example, an agent having a trust level of 1 for one provider and a trust level of 0.5 for another provider (on a scale from 0 to 1, with 0 being the lowest trust level), the agent would prefer to interact with the first one since the expected quality would be higher.

We can find a good number of trust models in the literature, reviews of which can be found in [11]. Trust models usually incorporate only direct and indirect trust interactions, as for example REGRET [13], which calls *individual dimension* the direct experiences and *social dimension* the indirect ones. This work introduces the ontological structure concept, that is, the reputation is considered as a combination of different features (for example, the reputation of an airline company can be composed of a set of features like delays, food quality, etc.) from where the overall reputation value is computed after assigning weights to the individual components. However, there are some approaches as in [1], where a trust model based just on a reputation mechanism is proposed. Other works, like [5], use three components of trust: direct, indirect and self. Conversely, FIRE [7] presents four different kinds of trust: "interaction trust" which is equivalent to direct trust, "witness reputation" which matches indirect trust, "certified reputation" which is similar to self trust besides the fact that it is not the agent itself who provides this trust but its partners (the agents more related to it), and finally "role-based trust" for the trust associated to the set of agents belonging to the same organization, and applied to all of its members.

Other differences between the models concern the method used to manage the information extracted from interaction, i.e. how the incoming information is added to compute the trust model or update it. Most of the methods use aggregation functions, as in [5, 19]. Others are based on probability theory [8, 17], information theory [14], or the Dempster-Shafer theory [18], among others. In order to work with dynamics (agents that change their services quality through time) and incomplete information some works use a *forgetting function* that enables the selection of the most relevant information to build the trust model, as in [5] or [7].

More focused on the ART domain (explained in the results section), we also find a large number of papers in which domain-dependent trust models and strategies have been designed with the aim of participating in different international competitions. In [9], as in many other ART agents, the agent design is divided in three parts: the strategy for modeling other agents (trust model), the requesting strategy and the response strategy. The winner of the 2006 and 2007 ART international competitions, known as IAM [16], consists of three main parts: a lie detector to detect malicious agents, a variance estimator to estimate the quality of the other agents, and an optimal weight calculator to measure its own behavior against others. A later improvement of the IAM agent presented a trust model based on Bayesian reinforcement learning [15]. Another technique that has been used to design ART agents in order to deal with untrustworthiness is fuzzy logic. In [2], fuzzy logic is used to normalize the information received from each agent.

3 Using the Trust Model

Although there is a lot of work about trust models, there is less about how to use them. So, here we present a strategy for using a trust model. The trust model can be any that provides the following features:

- The direct, indirect and self trust of the agents, for each of the services that they offer, with a normalizable value between 0 and 1.
- The knowledge degree of a provider agent about a service (based only on direct trust). This is a value (normalizable between 0 and 1) that represents how much the agent has directly interacted with a provider, with 0 meaning that the provider is totally unknown (the agent has never directly interacted with that provider), and 1 totally known.

The information that the trust model provides is used by the agent to take decisions regarding service requesting. It is important to note that even with a very complex model of trust and a good strategy, if the information contained in the model does not correspond to reality or is incomplete, the agent will not be able to take advantage of it. For example, an agent that does not know all of the agents will not be competitive against another agent that does know all of them, even if the latter has a worse trust model than the former.

The decision-making process involves the resolution of the exploration versus exploitation problem, given that the number of interactions to perform is usually limited. The key point here is the need to decide at every moment how much effort to spend on exploring for new unknown providers, and how much on exploiting the known ones.

In order to make this possible, our strategy is based on grouping the agents in four categories, according to the information stored about them in the trust model regarding a given service. Thus we consider the following categories:

- **Group TK (Totally Known agents):** The agents with a knowledge degree equal to 1. The trust model says that they are well-known providers since we have interacted with them many times.
- **Group PK (Partially Known agents):** Agents with a knowledge degree lower than 1 and greater than 0. These providers would probably become future providers in the event any of the existing ones failed.
- **Group AU (Almost Unknown agents):** Agents with a knowledge degree equal to 0. These are the agents without any direct interaction, but for which there is indirect or self information.
- **Group TU (Totally Unknown agents):** These are the agents without any information about, either direct, indirect or self.

Note that the membership of agents in the different groups changes through time, with all the agents belonging to the last group (TU) at the beginning. Moreover, agents that belong to the group TK may switch to lower groups if the trust model forgets old interactions (for dynamic environments).

These categories are used to define our exploration versus exploitation strategy. The total number of interactions T is limited due to time and cost restrictions, depending on the domain. Given a new set of services to be fulfilled, the agent exploits a set of M providers and explores a set of N agents, so that at the end the number of interacted agents is $T = M + N$.

3.1 Exploitation

In order to select the agents for exploitation, it is preferable to select from the providers belonging to the first group (TK), the ones that the agent thinks will provide good quality (direct trust higher than a given threshold QT, parameter that has to be set depending on the domain and the trust model) in supplying the requested service, as they are the most reliable agents. The services provided by these agents can be trusted since the agent has satisfactorily interacted with them many times in the past, and consequently it is quite likely that future interactions will be successful as well. In the case that there are not sufficient agents conforming to the above restriction, the agent can use as providers the rest of the agents from the first group, or the best agents from the second group (at a higher level of risk). As a result of the exploitation, the parameter M will be fixed (the number of agents with a direct trust higher than QT, with an upper limit of T).

3.2 Exploration

The main idea of the exploration process is to gradually raise agents from lower to higher groups until they conform to the first group's constraints. It is not mandatory to use exploration, but its benefits are clear, since without it, if the usual providers fail (the providers become unavailable or their quality decreases too much), the agent would have to start searching, perhaps interacting with totally unknown (potentially risky) agents in order to find new providers. Instead, choosing correctly the agents for exploration will move more agents to the known category, thus allowing the agent to have candidates for future exploitation, in the event that any regular providers are lost.

The exploration process could be performed by choosing the unknown agents randomly. However, we have designed a strategy that achieves a better outcome (as demonstrated in the results section 4.4). This mechanism consists of three phases and its objective is to optimally spread the exploration interactions of the agent. The agents to explore are taken from the groups PK, AU and TU, with the available interactions ($N = T - M$) being distributed according to the three following phases:

1. Select agents from the PK group. Agents are sorted in descending order according to their direct trust regarding the service to be covered, and only the best are selected. The objective of this phase is to know completely these agents, in order to move them to the TK group.
2. If there are still interactions left, the agents are taken from the AU group. The agents in this set are arranged in descending order according to their indirect trust, and only the best are selected. A single interaction is assigned to each provider, until exhausted. These agents will belong to the PK group in the next time step.
3. Finally, the agents are selected from the TU group, and a single interaction is performed in order to move them to the AU group. Here, the providers are selected at random, as we do not know anything about them.

The different phases of the mechanism are subsequently executed until the available exploration interactions (N) are exhausted.

3.3 Initial Time Steps Procedure

In the initial time steps no trust model has yet been built, and indirect trust is not useful either, because initially nobody knows anybody, so any received reputations would not be right. Therefore, the strategy here is to use the self trust values obtained from the agents to decide which agents to rely on.

Another strategy for the initial time steps is to trust the agents that answer our requests, and the agents that make requests to us. This is because we expect that agents interested in interacting with us will be truthful agents, while untruthful agents are generally less interested in interacting with others.

3.4 Agent Behavior for Service Providing

Up to now we have discussed how our agent requests services to others (requesting services). Here we talk about strategies for the agent behavior with the other agents' requests (providing services).

We believe that giving bad services qualities deliberately is not beneficial since in the long term the other agents either stop trusting, stop requesting, or even start giving bad qualities to us. Thus, we lose possible providers. Moreover, in dynamic environments, an agent that at a given time is not a good provider can become one in the future. Therefore, acting malevolently would produce a reaction against us in the agents, so we decide to act always honestly.

Finally, with regard to reputation requests, we decided to answer them truthfully. In doing so, the agents with good trust models (models that make use of indirect trust) are favored over the rest, and we expect them to answer back in the same way, thereby achieving mutual collaboration.

4 Experimentation

In this section we explain the tool used to test our work: the ART Testbed, and the results obtained on the ART Testbed international competition.

4.1 ART Testbed

The Agent Reputation and Trust Testbed[1] is a simulator for the art appraisal domain, "a working framework created with the goal of serving as an experimentation tool of different trust models and strategies for them, as well as a forum in which researchers can compare their technologies against objective metrics" [4]. It simulates an environment in which the participant agents act as art *appraisers* that compete to obtain the most clients. Appraisers receive more clients, and thus more profit, for giving more accurate appraisals. The simulator generates a fixed total number of clients that request appraisals for paintings that belong to different artistic eras (e.g. realism, impressionism, expressionism, etc.). The appraisers have varying levels of *expertise* in the different eras, making them experts in some but ignorant about others. This expertise can vary through time. During the game, the simulator progressively modifies the appraisers' client share according to the quality of their appraisals.

For the agents to perform the appraisals, they can ask other agents for their opinions (especially for the eras in which they are not experts) and set a weight for each of them in the simulator (expected to correspond to the agents trust). The final appraisal is then computed by the simulator as the mean of the agents' weighted appraisals. Each kind of interaction implies a payment from the requesting agent to the requested agent of a given amount of money, independent of the quality obtained in the interaction. The payment costs are more expensive for direct interactions than for other kinds of interactions. When the agent receives

[1] http://www.art-testbed.net

opinion requests from other agents, it has to decide how much money to spend on the opinion (the quality of the opinion depends on the amount of money spent). This money represents the appraisal effort made by the agent.

In addition to conducting opinion transactions (direct trust), appraisers can exchange *reputations*, or information about the trustworthiness of other appraisers (indirect trust). The self trust in this domain is called *certainty*. An important feature of the ART Testbed is that it is not necessary for the agents to provide good information. For each agent's request they can decide whether to cooperate or to act maliciously.

4.2 ART Agent

Figure 1 shows a simplified diagram of the interactions of our agent within the ART Testbed simulator (only direct interactions are drawn). First, the simulator sends the paintings to the agents (appraisal requests). The agent has to decide which agents to request an opinion from about each painting. As a result, the agent returns back the simulator a weight for each agent and era. The weight for an agent x and era j represents the importance that the agent wants to give to agent x's opinion for the paintings appraised of era j. Finally, the simulator calculates the final appraisal of the paintings and performs the weighted sum (with the weight given by the agent) of the appraisals requested. At the following time step, the agent will know the result of the appraisals given by other agents since the simulator sends the true values of the paintings.

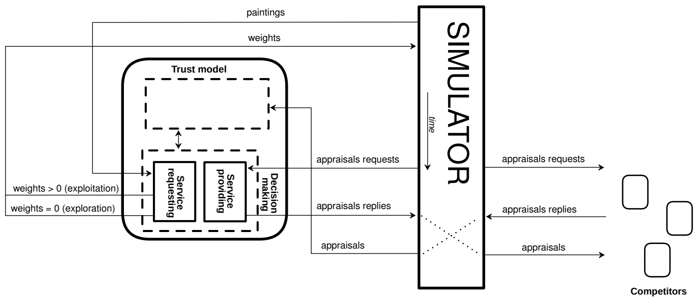

Fig. 1. Interactions with the ART Testbed simulator

In this domain the agent can simulate exploration and exploitation with the weights, by giving a weight greater than zero for exploitation, and equal to zero for exploration. When a weight equal to zero is given to the simulator the request is performed, so the agent will later know the appraisal given by agent but this will not affect the final appraisal.

The amount of effort dedicated to exploitation (M interactions) is given by the quality threshold QT, so that agents with a trust value higher than this threshold will be used as appraisers, with a limit of T. This defines the maximum number of questions that the agent wants to perform for each painting in order to spend a given percentage ($questionPercentage$) of the earnings of an opinion ($\frac{clientFee}{opinionCost}$). This limit is calculated according to Equation 1:

$$T = \min\left(\frac{clientFee \cdot questionPercentage}{opinionCost}, maxNbOpinionRequests\right) \quad (1)$$

where $clientFee$, $opinionCost$ and $maxNbOpinionRequest$ are parameters of the simulator representing, respectively, the price earned at each transaction, the cost of a requested opinion, and the maximum number of opinion requests that an agent can send for each painting at each time step. Alternatively, the parameter $questionPercentage$ is agent's own and indicates the percentage of the earnings that it wants to spend on asking other agents.

During the exploitation selection M interactions have been performed, if there are still interactions left (if $M < T$), the rest ($N = T - M$) will be used in the exploration process, following the phases previously explained in Section 3.

4.3 Exploration Algorithm Evaluation

With the aim of evaluating our exploration procedure for the selection of agents to be discovered in the ART Testbed domain, the following experiment was designed. We created a game with 10 copies of our agent equipped with the trust-based exploration mechanism previously explained. We also added to the game ten copies of the agent with a random selection of candidates to explore. The trust model, the exploitation mechanism and the parameters were the same for the two groups of agents. We also added dummy agents to the game: five dishonest (act malevolently) and five honest. The parameters of the game were the following: 100 time steps, 20 eras, 4 $numberOfErasToChange$. The rest of the parameters were set to their default values.

In order to obtain a metric for the comparison we defined the useful information index (UII), which measures the useful information that the agent has found in the environment. This metric is defined as follows:

$$UII = \frac{\sum_{i=0}^{i<numEras} GA(i)}{numEras} \quad (2)$$

where the value of $GA(i)$ is 1 when the agent has two or more agents with a quality in the era i higher than the quality threshold (QT). The value is 0.5 if there is only one agent satisfying this condition and 0 if there are no agents.

Figure 2 shows the UII average of the ten agents with the trust-based exploration mechanism and the UII average of the ten agents with the random mechanism. We can see that a stabilization phase appears during the 20 first time steps. During this phase the agents get to know the environment until they arrive at a stabilization point. From that point, the UII value oscillates due to the expertise changes produced in the environment. During the stabilization phase, the UII grows faster and reaches a higher maximum value in the agents with

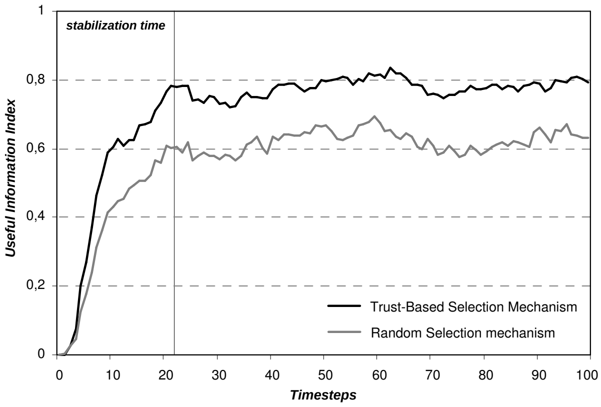

Fig. 2. Comparison between the trust-based exploration mechanism and a random exploration mechanism

the trust-based exploration strategy than with the random mechanism. Furthermore, the value reached can be maintained at this higher value. Therefore, we conclude that the designed strategy allows the agent to feed the trust model with more useful information than a random method.

4.4 Competition Results

We now analyze the performance of the agent compared with other real agents. We designed an agent named Uno2008 that uses a trust model similar to [5] with some adjustments. Due to space limitations we cannot explain the whole trust model (for a detailed explanation see [10]). The parameter QT has been set to 0.7, and $questionPercentage = 0.4$; these values have been found empirically, although the behavior of the agent does not change abruptly with similar values.

The results are taken from the 2008 International ART Testbed Competition held in Estoril, at AAMAS. In this competition 11 agents were registered from 7 different countries. Five dummy agents were also added. The competition consisted of three different kinds of games, the first with low dynamics (# eras to change (ETC) = 1, amount of expertise change (AEC) = 0.05), the second with medium (# ETC = 3, AEC = 0.1) and the third with high dynamics (# ETC = 3, AEC = 0.3). Each was repeated three times, and the average of earnings of the agents in the three games was computed to determine the final scores.

The results are shown in Figure 3, where the y axis represents the average earnings obtained for each agent in all the games, with its standard deviation. Our agent, Uno2008, managed to win the competition by obtaining the highest score, with Connected and FordPrefect completing the podium. Agent Uno2008 won eight of the nine games played. We must also highlight the big difference in the first four players over the rest. The last five agents, called “seimmud”, correspond to the dummy agents.

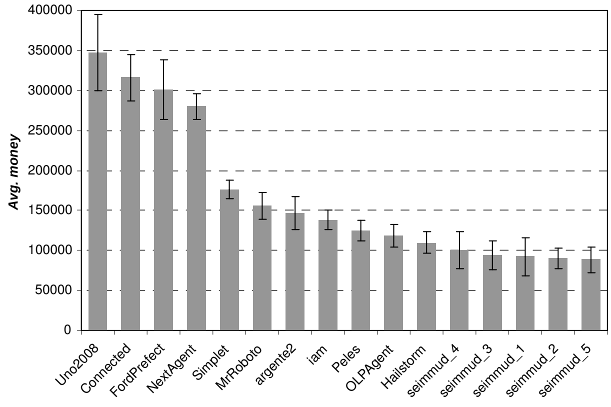

Fig. 3. Average and standard deviation of the earnings in the 2008 ART Competition

5 Conclusions

In environments with competitive agents, an agent can behave maliciously, trying to harm other agents, with the aim of obtaining better results. Alternatively, the agents can offer different service qualities. For these reasons, trust is a very important factor as it allows us to know about the behavior of agents and to predict the results of interactions with them, and consequently to make better decisions. However, a perfect trust model is difficult to obtain, and therefore the design of a strategy to take correct decisions based on an incomplete trust model (which may contain mistakes) is also an important factor if the agent is to obtain maximum benefits.

In this article, a strategy for using a trust model in a decision-making process has been presented. The required trust model must be based on three different trust components: direct, indirect and self. Direct trust is based on the agent's own experiences, indirect trust (reputation) is based on other agent's experiences, and self trust is the publicity that an agent transmits about itself. The data of an agent's trust model has to be processed in order for the best decisions to be taken and leading to its benefits being maximized or its objectives being obtained. The process of making the decisions involves the exploitation versus exploration problem. To solve this problem, we classify the agents in four categories (totally known, partially known, almost unknown and totally unknown). We use a method for exploration that combines the chance of finding good information in partially known agents with the random factor.

The design of the agent was tested in the ART Testbed domain. It participated in the 2008 international competition held in Estoril (in AAMAS), which it won. As future work, we are studying the possible application of these strategies in other real domains, such as Internet electronic service providers, to check whether they behave as well as they did in the ART competition.

Acknowledgments. This research project has been partially funded by the Spanish MICINN project SuRoS (TIN2008-04547) and with the support of the Commissioner for Universities and Research of the Department of Innovation, Universities and Company of Generalitat of Catalonia and of the European Social Background, and the University of Girona BR Grant program.

References

1. Abdul-Rahman, S.H. A., Supporting trust in virtual communities, in the 33rd IEEE International Conference on Systems Sciences, 4-7, (2000).
2. Del Acebo, E., Hormazábal, N., and de la Rosa, P., Beyond Trust. Fuzzy Contextual Corrective Filters for Reliability Assessment in MAS, The Workshop on Trust in Agent Societies at AAMAS-2007, Honolulu, Hawaii, USA, May 15, pp. 44-48, (2007).
3. Falcone R., Singh M.P. and Tan Y., Trust in Cyber-societies, Integrating the Human and Artificial Perspectives, Lecture Notes in Computer Science, 2246, 54-72 (2001).
4. Fullam K., Sabater J. and Barber S., Toward a testbed for trust and reputation models, Trusting Agents for Trusting Electronic Societies, 95-109, (2005).
5. Gómez M., Carbó J., Benac C., Honesty and trust revisited: the advantages of being neutral about other's cognitive models, Autonomous Agent and Multi-Agent Systems, 15, 313-335 (2007).
6. Gui, C., Wu, Q., Wang, H., Towards Trustworthy Resource Selection: A Fuzzy Reputation Aggregation Approach. Autonomic and Trusted Computing, 4610/2007, 239-248, (2007).
7. Huynh, T.D., Jennings N., Shabdolt N.R., An integrated trust and reputation model for open multi-agent systems, AAMAS 2005, 512-518, (2005).
8. Josang, A., Ismail, R., The beta reputation system, in proceedings of the 15th Bled Conference on Electronic Commerce, (2002).
9. Kafali, O., Yolum, P., Trust Strategies for the ART Testbed, the Workshop on Trust in Agent Societies at AAMAS-2006, 43-49, (2006).
10. Muñoz, V., Murillo, J., Agent UNO: Winner in the 2nd Spanish ART competition, Revista Iberoamericana de Inteligencia Artificial, 39, 19-27, (2008).
11. Ramchurn, S., Hyunh. D., Jennings, N.R., Trust in multi-agent systems, The Knowledge Engineering Review, 19, 1-25, (2004).
12. Sabater, J., Trust and Reputation for Agent Societies, Monografies de l'institut d'investigació en intel.ligència artificial, 20, PhD Thesis, (2003).
13. Sabater, J., Sierra C., Regret: A reputation model for gregarious societies, in Fourth Workshop of Deception, Fraud and Trust in Agent Societies, 61-69, (2001).
14. Sierra, C., Debenham, J., An information-based model of trust, AAMAS 2005, 497-504, (2005).
15. Teacy, W.T.L., Chalkiadakis, G., Rogers, A., Jennings, N. R., Sequential Decision Making with Untrustworthy Service Providers, AAMAS 2008, 755-762, (2008).
16. Teacy, L., Huynh, T.D., Dash, R.K., Jennings, N.R., Luck, M., Patel, J., The ART of IAM: The Winning Strategy for the 2006 Competition, Proceedings of the 10th International Workshop on Trust in Agent Societies, 102-111, (2007).
17. Teacy, W., Patel, J., Jennings, N., Luck, M., TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources, AAMAS, 12, 183-198, (2006).
18. Yu, B., Singh M., A social mechanism for reputation management in electronic communities, Lecture notes in computer science, 1860, 154-165, (2000).
19. Zacharia, G., Maes P., Trust management through reputation mechanisms, Applied Artificial Intelligence, 14, 881-907, (2000).