A Maintenance-based trust for Open MultiAgent Systems

Khosravifar, Babak; Gomrokshi, M; Bentahar, Jamal; Thiran, Philippe

Published in:
Proc. of the 8th International Conference on Autonomous Agents and Multiagent Systems

Publication date:
2009

Maintenance-based Trust for Multi-Agent Systems

Babak Khosravifar
Dep. of Elec. and Comp. Eng.
Concordia University, Canada
b_khosr@encs.concordia.ca

Maziar Gomrokchi
Dep. of Comp. Science
Concordia University, Canada
m_gomrok@encs.concordia.ca

Jamal Bentahar
Concordia Inst. for Inf. Sys. Eng.
Concordia University, Canada
bentahar@ciise.concordia.ca

Philippe Thiran
University of Namur
Namur, Belgium
pthiran@fundp.ac.be

ABSTRACT

In last years, trust and reputation has been gaining increasing interest in multi-agent systems (MAS). To address this issue, we propose in this paper a maintenance-based trust mechanism for agents operating in multi-agent systems. In the proposed model, a comprehensive trust assessment process is provided to assess the trustworthiness of the participating agents. The main characteristic of this model is the retrospect trust adjustments, which integrate the applicable constraints and modify the involved features with respect to the actual performance of the evaluated agent. Specifically, the retrospect process updates the belief set of the agents in order to adapt them to the social network changes. This paper has two contributions: after describing the architecture of the proposed framework, we provide a theoretical analysis of its assessment and discuss the system implementation, along with simulations comparing it with the broadly known frameworks.

Keywords. Trust, Multi-Agent Systems, Agent Communication.

1. INTRODUCTION

Over the last recent years, agent communication protocols have been well established in MAS. In such systems, autonomous agents are distributed in large-scale network and interact to collaborate and share resources with each other. Trust is essential in such settings to provide a social control in effective interactions [1, 7]. Generally, an agent's trust in another is defined as the measure of willingness that the agent will fulfill what he agrees to do and computed by considering personal interaction experiences and collecting suggested ratings from others. In such distributed systems, the computed trust enables agents to reason about the likely intentions of others that are not known and thus assess the trustworthiness of the interacting agents.

To maintain a trust-based network, different computational frameworks have been proposed in the literature. Each of those proposed models addresses some features that may distract the trust assessment efficiency. Some models consider the direct interaction of two parties [4, 6]. Some models rely, to some extent, on the suggested rating provided by other agents [10, 11, 12]; and some others also consider the suggested rating of the agent being evaluated [1, 4]. Since agents are self-interested, it is hard to analyze an agent's likely behavior based on previous direct interactions [2] given the fact that the collected information from other agents may be non-reliable and could lead to a non-accurate trust assessment. So far, these frameworks do not act properly if selfish agents tend to change their behaviors. Therefore, they do not recognize the recent improvement or degradation in particular agent's capabilities. Considering these limitations, the trust models aim to act more efficiently in terms of assessment accuracy and to be adaptive to the environment inconsistencies.

In this paper, we propose a model aiming to advance results obtained by existing trust frameworks in the literature. We provide an efficient assessment process in a twofold contribution. In the first contribution, agents mutually interact and rate each other based on the interaction done (satisfactory or dissatisfactory). The obtained ratings are accumulated to assess the direct interaction rating of a particular agent. Inter-agent communication is regulated by protocols and determined by strategies. Upon evaluating an unknown or not very well-known agent (we call this agent *the trustee* and refer to him as Ag_b), the evaluator agent (we call this agent *the trustor* and refer to him as Ag_a) is able to ask others (*consulting agents*) about their direct interaction rating with the trustee agent. The consulting agents are composed of *trustworthy agents* (known by the trustor agent) and *referee agents* (introduced by the trustee agent). In the proposed framework, Ag_a evaluates the credibility of Ag_b by combining his own direct trust rating with the ratings provided by the consulting agents. The suggestions provided by these agents are partially considered with respect to their time recency, interaction strengthen and accuracy. In the second contribution of this paper, the trustor agent after a period of direct interaction with the trustee agent performs a retrospect trust adjustment (so called maintenance) in order to update his belief set about the credibility of the consulting agents (trustworthy and referee agents) that provided information regarding to the trust level of trustee agents. In the

Cite as: Maintenance-based Trust for Multi-Agent Systems, Babak Khosravifar, Maziar Gomrokchi, Jamal Bentahar, Philippe Thiran, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1017–1024

periodic maintenance process, the suggestions provided by consulting agents are compared with the observed behavior of the trustee agents. Exceeding some predefined thresholds, the trustor agent would either increase or decrease his trust ratings about consulting agents. Doing so, gradually agents recognize more reliable consulting agents around in the network, which would cause a more efficient trust assessment process in future. This assessment could be used in large scale social networks and new generation of web services. In this paper, we analyze the effect of the maintenance process in different points of view and we compare the system efficiency with some other models.

The remainder of this paper is organized as follows. In Section 2, we define our framework as comprehensive trust assessment process, which is composed of direct and indirect evaluation process. In Section 3, we discuss the maintenance that a typical agent makes after a certain period, since the interactions initiated. In Section 4, we outline the properties of our model in the experimental environment. Representing the testbed, we compare our model results with two well-known trust models in terms of efficiency in trust assessment. Section 5 discusses related work and finally Section 6 concludes the paper.

2. COMPREHENSIVE TRUST ASSESSMENT

In this section, we discuss the comprehensive trust assessment process, which the trustor agent Ag_a performs for estimating the credibility of the trustee agent Ag_b. The evaluation is a twofold approach: direct and indirect. In the former approach, Ag_a only relies on his previous interactions with Ag_b. The previous interactions affect the assessment process depending on their quantity (number of interactions) and freshness (time recency). In the later approach, in addition to the direct interactions, Ag_a also consults some other agents to assess the credibility of Ag_b. A number of consulting agents are selected and their credibility and the coherence of the information they provide are to be analyzed.

2.1 Direct Trust Evaluation

If in the social network, agents know each other, this means that they had prior interaction history and thus can directly compute the trust value of each other. Using their reasoning capabilities, agents evaluate the outcomes of their interactions. In the general case, they can evaluate their interaction outcomes according to a scale of n types numbered from 1 (the most successful interaction) to n (the less successful interaction), such that the first m interaction types ($m < n$) are considered successful. Let $NI_i{}^{Ag_b}_{Ag_a}$ be the number of interactions of type i that the trustor Ag_a had with the trustee Ag_b. Then, the trustor's estimated trust value for the trustee, $DTr^{Ag_b}_{Ag_a}$ can be computed by equation 1 as the ratio of the "*number of successful outcomes*" to the "*total number of possible outcomes*":

$$DTr^{Ag_b}_{Ag_a}=\frac{\sum_{i=1}^{m}(w_i\times TiR(\Delta t^{Ag_b}_{Ag_a})\times\sum_{j=1}^{NI_i{}^{Ag_b}_{Ag_a}} v_{ij})}{\sum_{i=1}^{n}(w_i\times TiR(\Delta t^{Ag_b}_{Ag_a})\times\sum_{j=1}^{NI_i{}^{Ag_b}_{Ag_a}} v_{ij})} \quad (1)$$

where w_i is the weight associated to the interaction type i and the v_{ij} is the measure considered to avoid two transactions with different values being treated equally. In addition to the wight and measure of the interactions, another factor is used to reflect the *timely relevance* of transmitted information. Since agents act in an environment, which is dynamic and may change quickly, it is more desirable to promote recent information and deal with out-of-date information with less emphasis. In our model, we assess this factor denoted by $TiR(\Delta t^{Ag_b}_{Ag_a})$ by using the timely relevance function defined in equation 2.

$$TiR(\Delta t^{Ag_b}_{Ag_a}) = e^{-\lambda \; ln(\Delta t^{Ag_b}_{Ag_a})} \quad \lambda \geq 0 \quad (2)$$

$\Delta t^{Ag_b}_{Ag_a}$ is the time difference between the current time and the time at which Ag_a updates his information about Ag_b's trust. The intuition behind this formula is to use a function decreasing with the time difference. Consequently, the more recent the information is, the higher the timely relevance coefficient would be. Variable λ is an application-dependent coefficient. In some applications, recent interactions are more desirable to be considered. In that case the trustor uses a higher value for λ to judge the credibility of the trustee. In contrast, in some other applications, even the old interactions are still valuable source of information. In that case, the trustor sets a relatively smaller value to λ.

2.2 Combined Direct and Indirect Evaluation

The second approach in comprehensive trust estimation of the trustee agent is to collect information in terms of suggestion from some other agents (referred as consulting agents). As described before, the consulting agents are divided into two groups: (1) the trustworthy agents, that the trustor agent Ag_a can rely on to request for information; and (2) the referee agents, that are introduced by the trustee agent Ag_b as recommenders. In this section, we address the selection process of the consulting agents and how to deal with the information they provide in support of Ag_b.

Let $\mathcal{T}^{Ag_b}_{Ag_a}$ be the set of trustworthy agents that Ag_a knows from his belief set, which can report on Ag_b. Depending on the situation, how much Ag_a is aware of his surrounding environment and how restrictive Ag_a needs to be in the selection of consulting agents, a trustworthy selection threshold (μ_T) is supposed to set in order to select a required number of trustworthy agents and fill the selected trustworthy set $\mathcal{T}s^{Ag_b}_{Ag_a}$ (a typical element of this set is denoted by Ag_t). Basically, the elements of this set are the agents that are going to be asked about the credibility of the trustee Ag_b.

$$\mathcal{T}s^{Ag_b}_{Ag_a} = \{Ag_t \in \mathcal{T}^{Ag_b}_{Ag_a} | Tr^{Ag_t}_{Ag_a} > \mu_T\}$$

Another set to be involved in the evaluation process is the set of referee agents, which are introduced by Ag_b. Upon request from Ag_a, Ag_b replies by providing a list of the referee agents that he knows ($\mathcal{R}^{Ag_b}_{Ag_a}$). Following the restriction policy by the predefined threshold μ_R, Ag_a consequently selects the appropriate referee agents ($\mathcal{R}s^{Ag_b}_{Ag_a}$). The elements of $\mathcal{R}s^{Ag_b}_{Ag_a}$ (denoted by Ag_r) are the selected referee agents that Ag_a would consider their suggestions about Ag_b.

$$\mathcal{R}s^{Ag_b}_{Ag_a} = \{Ag_r \in \mathcal{R}^{Ag_b}_{Ag_a} | Tr^{Ag_r}_{Ag_a} > \mu_R\}$$

However, there are some other referees that are introduced by Ag_b but because of being unreliable or unknown, they have not been asked about Ag_b.

$$\mathcal{R}s'^{Ag_b}_{Ag_a} = \mathcal{R}^{Ag_b}_{Ag_a} - \mathcal{R}s^{Ag_b}_{Ag_a}$$

In this case Ag_a does not consider these agents' suggestions about Ag_b, but he saves the referees' suggestion in order to compare by the real behavior Ag_b performs after starting interaction with Ag_a. This comparison is made in the retrospect process, that we discuss in depth in Section 3. After comparison, such referee agents are known by Ag_a and their trust levels are computed by the adjustment of what they provided and Ag_b's real behavior. After selecting the proper consulting agents to ask about Ag_b, Ag_a asks each one of them about the rating they can provide. In the proposed framework, depending on some restriction factors, the obtained suggestions are partially considered in the total trust evaluation. The first restriction factor is the time recency, which the affect is discussed in section 2.1 and derived from equation 2. The second restriction factor is denoted as the *relation strengthen*. Let $NI_{Ag_x}^{Ag_y}$ be the total number of interactions between two agents Ag_x and Ag_y, which is computed by the equation 3, where n is a number of interaction types (see equation 1).

$$NI_{Ag_x}^{Ag_y} = \sum_{i=1}^{n} {NI_i}_{Ag_x}^{Ag_y} \tag{3}$$

It is worth to mention that the total number of interactions between Ag_t (resp. Ag_r) and Ag_b, $NI_{Ag_t}^{Ag_b}$ (resp. $NI_{Ag_r}^{Ag_b}$) is an important factor because it promotes information obtained from agents that are more acquainted with Ag_b. Generally, these agents are considered as good sources of information because of their high number of interactions with Ag_b. The third restriction factor considered by our trust model is the trustworthiness of the consulting agents. Considering his belief, Ag_a assigns a trust value for each consulting agent and thus restricts their contribution with respect to their relative trust value.

The trustor Ag_a derives the total trust estimation of Ag_b by taking into account the aforementioned restriction factors, which are categorized as follows: (1) the trustworthiness of trustworthy/referee agents from Ag_a's point of view ($Tr_{Ag_a}^{Ag_t}/Tr_{Ag_a}^{Ag_r}$); (2) Ag_b's trustworthiness according to the point of view of trustworthy/referee agents ($Tr_{Ag_t}^{Ag_b}/Tr_{Ag_r}^{Ag_b}$); (3) the total number of interactions between trustworthy/referee agents and Ag_b ($NI_{Ag_b}^{Ag_t}/NI_{Ag_b}^{Ag_r}$); and (4) the timely relevance of interactions between trustworthy/referee agents and Ag_b ($TiR(\Delta t_{Ag_t}^{Ag_b})/TiR(\Delta t_{Ag_r}^{Ag_b})$). Before giving the equation computing this trust estimation, we need to introduce the functions Ω_T, Ω_R, Ω'_T and Ω'_R, that are used to assess the trustee's trust value by combining the trust values estimated by the trustworthy and referee agents. Let A be a set of agents. Formally, these functions are defined as follows:

$$\Omega_T, \Omega'_T, \Omega_R, \Omega'_R : 2^A \rightarrow \mathbb{R}^+$$

Because of space limit, we only define the measure functions Ω_T and Ω'_T, which are related to the trustworthy agents. Ω_R and Ω'_R (related to referee agents) are set likewise. The equation computing the trust estimation ($Tr_{Ag_a}^{Ag_b}$) is given by equations 4, 5 and 6.

$$Tr_{Ag_a}^{Ag_b} = \frac{\Omega_T(\mathcal{T}s_{Ag_a}^{Ag_b}) + \Omega_R(\mathcal{R}s_{Ag_a}^{Ag_b})}{\Omega'_T(\mathcal{T}s_{Ag_a}^{Ag_b}) + \Omega'_R(\mathcal{R}s_{Ag_a}^{Ag_b})} \tag{4}$$

$$\Omega_T(\mathcal{T}s_{Ag_a}^{Ag_b}) = \sum_{Ag_t \in \mathcal{T}s_{Ag_a}^{Ag_b}} Tr_{Ag_a}^{Ag_t} \times Tr_{Ag_t}^{Ag_b} \times TiR(\Delta t_{Ag_t}^{Ag_b}) \times NI_{Ag_b}^{Ag_t} \tag{5}$$

$$\Omega'_T(\mathcal{T}s_{Ag_a}^{Ag_b}) = \sum_{Ag_t \in \mathcal{T}s_{Ag_a}^{Ag_b}} Tr_{Ag_a}^{Ag_t} \times TiR(\Delta t_{Ag_t}^{Ag_b}) \times NI_{Ag_b}^{Ag_t} \tag{6}$$

Following the ideology that Ag_a could, to some extent, rely on his own history interaction with Ag_b (direct trust evaluation approach) and partially use the second approach, which is consulting other agents, Ag_a gives a %100 trustworthy rate ($Tr_{Ag_a}^{Ag_a} = 1$) to his history and considers himself as a member of his trustworthy community. By so doing, equation 4 combines direct and indirect approach in the same formula. This merging method takes into account the proportional relevance of each trust assessment, rather than treating them separately. Basically, the contribution percentage of each approach in the final evaluation of $Tr_{Ag_a}^{Ag_b}$ is defined regarding to how informative the history is in terms of the number of direct interactions between Ag_a and Ag_b and their time recency. Consequently, consultation with other agents is less considered if the history represents a lower entropy, which reflects lower uncertainty. The lower entropy of the history means that it is more informative and thus reliable. Respectively, the higher entropy of the history makes the trustor uncertain and thus rely less on that history. Therefore, consultation with other agents should be considered. To be more precise, we analyze the quality of the interactions with the trustee agent regarding to what is expected (trust evaluation $Tr_{Ag_a}^{Ag_b}$) and what is actually performed (so-called observed trust value $OTr_{Ag_a}^{Ag_b}$). To this end, we propose a retrospect trust evaluation algorithm, which is represented in Section 3.

3. RETROSPECT TRUST ADJUSTMENT

3.1 Adjustment as Optimization Problem

Generally, in dynamic MAS, interacting agents change their behaviors and behave regarding to their new intentions. It is crucial in such a setting that agents adapt themselves with the environment inconsistencies. To this end, we provide a mechanism that periodically performs a maintenance process to efficiently adapt with the environment changes. In this mechanism, the trustor agent adjusts his belief about consulting agents (trustworthy and referee agents) that (in a particular period) was involved in one or few trust assessments the trustor performs before interaction with any trustee agent. The belief set is updated considering the overall accuracy of the consulting agents in providing information. In so doing, after each interaction with any trustee agent Ag_b, the trustor Ag_a would record the suggestions provided by the consulting agents. Afterwards, comparing the estimated trust value with the observed behavior of the trustee, the trustor analyzes the possible updates in the partial ratings that he could allocate to the consulting agents in order to decrease the difference between the estimated trust and the observed behavior. To clarify this process we define some parameters in the following paragraphs.

Consider a particular trust assessment process performed by Ag_a before interaction with Ag_b as the trustee. In this process, $\mathcal{T}s_{Ag_a}^{Ag_b}$ and $\mathcal{R}s_{Ag_a}^{Ag_b}$ respectively represent the set of trustworthy and referee agents involved in that process. Let $\mathcal{C}s_{Ag_a}^{Ag_b}$ be the set of involved consulting agents ($\mathcal{C}s_{Ag_a}^{Ag_b} = \mathcal{T}s_{Ag_a}^{Ag_b} \bigcup \mathcal{R}s_{Ag_a}^{Ag_b}$). We refer to the trust value $Tr_{Ag_a}^{Ag_i}$ given by Ag_a to a consulting agent Ag_i ($Ag_i \in \mathcal{C}s_{Ag_a}^{Ag_b}$) as the given rate. Hence, the set of rates given by Ag_a to all the consulting agents in that particular process is represented by the vector $\underline{\mathcal{T}r}_{Ag_a}^{Ag_b}$. The given rates together with the suggested values $Tr_{Ag_i}^{Ag_b}$ and the supplementary information ($NI_{Ag_i}^{Ag_b}$ and $TiR(\Delta t_{Ag_b}^{Ag_i})$ in equation 4) are used to compute $Tr_{Ag_a}^{Ag_b}$ as the trust estimation of Ag_a for Ag_b and after the interaction, $OTr_{Ag_a}^{Ag_b}$ is refereed as the actual behavior of Ag_b observed by Ag_a. Here the challenge is how Ag_a can update his belief set to give more appropriate rates to the consulting agents that upon trust evaluation process, achieve the highest accuracy. Basically, Ag_a seeks for a set of ratings that for any trustee agent could minimize the difference between the estimated trust and the observed behavior.

$$min \ \ |Tr_{Ag_a}^{Ag_b} - OTr_{Ag_a}^{Ag_b}|$$

In order to achieve the minimized difference, the trustor agent Ag_a, comparing the suggested values with the observed value of Ag_b, builds up a set of constraints, which are used to compute updated given ratings vector $\underline{\mathcal{UPT}r}_{Ag_a}^{Ag_b}$. The elements of this vector (denoted by $\mathcal{UPT}r_{Ag_a}^{Ag_i}$ where $Ag_i \in \mathcal{C}s_{Ag_a}^{Ag_b}$) represent the updated trust value for each consulting agent that participated in the trust assessment process. Basically, the constraints are used to restrict the answers that could be obtained as a result of estimation error minimization. Refusing to set up the appropriate constraints can lead to some inconsistencies in the sense that the updated ratings overestimate (or underestimate) the consulting agents. This may cause loss of incentive for the consulting agents to provide accurate information in future. Hence, we set up these constraints particulary in two perspectives: (1) any consulting agent that provided trust rate for Ag_b within acceptable range of accuracy error (ν) should eventually obtain an increase in his trust rate given by Ag_a and in contrast, any consulting agent that provided trust rate outbound the accuracy error is subject to be penalized by decreasing his trust rate (the value ν is set depending on how restrictive Ag_a is); and (2) any consulting agent Ag_i with respect to the information entropy he has (dependent of $NI_{Ag_i}^{Ag_b}$ and $TiR(\Delta t_{Ag_i}^{Ag_b})$) and the provided information accuracy error ($D_{Ag_i} = |Tr_{Ag_i}^{Ag_b} - OTr_{Ag_a}^{Ag_b}|$) obtains either a scaling up rate (SUR_{Ag_i}) or scaling down rate (SDR_{Ag_i}). $\underline{SUR}$ and $\underline{SDR}$ are the vectors representing these values for all consulting agents. These rates relatively show how important a consulting agent can be for Ag_a. Figure 1 represents the algorithm that builds the aforementioned constraints.

We respect the fact that consulting agents that had high number of interactions and time recency should provide more accurate information. By doing so, the consulting agents are divided into two sets: $SU_{Ag_a}^{Ag_b}$ and $SD_{Ag_a}^{Ag_b}$. The set $SU_{Ag_a}^{Ag_b}$ contains the consulting agents Ag_i having an accuracy error D_{Ag_i} less than the error threshold ν and in contrast the set $SD_{Ag_a}^{Ag_b}$ contains those agents that have larger accuracy error. Following the first perspective of the constraints, for each agent of the set $SU_{Ag_a}^{Ag_b}$, we assign a constraint in the sense that he does not loose his given trust rate by Ag_a. Likewise, the corresponding set of constraints are assigned for the agents belonging to the set $SD_{Ag_a}^{Ag_b}$. We accumulate the constraints in the set $\mathcal{C}ons_{Ag_a}^{Ag_b}$ (constraints are formulated as mathematical inequations). Following the second perspective of the constraints, the scaling rates ($\underline{SUR}$ and $\underline{SDR}$) are sorted. The assigned constraints reflect the property that the rates of increase in agents belonging to $SU_{Ag_a}^{Ag_b}$ (and the rates of decrease in agents belonging to $SD_{Ag_a}^{Ag_b}$) depend on the proportion regarding to their scaling rate values.

function addConstraints($Ag_a, Ag_b, \mathcal{C}s_{Ag_a}^{Ag_b}, Tr_{Ag_a}^{Ag_b}, OTr_{Ag_a}^{Ag_b}$)
  $\mathcal{C}ons_{Ag_a}^{Ag_b} = \{\ \}$; $SU_{Ag_a}^{Ag_b} = \{\ \}$; $SD_{Ag_a}^{Ag_b} = \{\ \}$;
  for all $Ag_i \in \mathcal{C}s_{Ag_a}^{Ag_b}$
   $D_{Ag_i} = |Tr_{Ag_i}^{Ag_b} - OTr_{Ag_a}^{Ag_b}|$;
   if ($D_{Ag_i} < \nu$)
    $SU_{Ag_a}^{Ag_b} = SU_{Ag_a}^{Ag_b} \bigcup \{Ag_i\}$;
    $\mathcal{C}ons_{Ag_a}^{Ag_b} = \mathcal{C}ons_{Ag_a}^{Ag_b} \bigcup \{1 > \mathcal{UPT}r_{Ag_a}^{Ag_i} \geq Tr_{Ag_a}^{Ag_i}\}$;
    $SUR_{Ag_i} = \frac{NI_{Ag_i}^{Ag_b} \times TiR(\Delta t_{Ag_i}^{Ag_b})}{D_{Ag_i}}$;
    add SUR_{Ag_i} *to* $\underline{SUR}$;
   else
    $SD_{Ag_a}^{Ag_b} = SD_{Ag_a}^{Ag_b} \bigcup \{Ag_i\}$;
    $\mathcal{C}ons_{Ag_a}^{Ag_b} = \mathcal{C}ons_{Ag_a}^{Ag_b} \bigcup \{0 < \mathcal{UPT}r_{Ag_a}^{Ag_i} \leq Tr_{Ag_a}^{Ag_i}\}$;
    $SDR_{Ag_i} = \frac{NI_{Ag_i}^{Ag_b} \times TiR(\Delta t_{Ag_i}^{Ag_b})}{D_{Ag_i}}$;
    add SDR_{Ag_i} *to* $\underline{SDR}$;

  sort($\underline{SUR}$);
  sort($\underline{SDR}$);
  for $i = 1$ to $|\underline{SUR}| - 1$
   $newConstraint =$
$\{\mathcal{UPT}r_{Ag_a}^{Ag_i} - Tr_{Ag_a}^{Ag_i} \geq \frac{SUR_{Ag_i}}{SUR_{Ag_{i+1}}}(\mathcal{UPT}r_{Ag_a}^{Ag_{i+1}} - Tr_{Ag_a}^{Ag_{i+1}})\}$;
   $\mathcal{C}ons_{Ag_a}^{Ag_b} = \mathcal{C}ons_{Ag_a}^{Ag_b} \bigcup newConstraint$;
  for $i = 1$ to $|\underline{SDR}| - 1$
   $newConstraint =$
$\{Tr_{Ag_a}^{Ag_i} - \mathcal{UPT}r_{Ag_a}^{Ag_i} \geq \frac{SUR_{Ag_i}}{SUR_{Ag_{i+1}}}(Tr_{Ag_a}^{Ag_{i+1}} - \mathcal{UPT}r_{Ag_a}^{Ag_{i+1}})\}$;
   $\mathcal{C}ons_{Ag_a}^{Ag_b} = \mathcal{C}ons_{Ag_a}^{Ag_b} \bigcup newConstraint$;

Figure 1: Constraint making algorithm for updating trust rating, performed by agent Ag_a.

After defining the set of constraints $\mathcal{C}ons_{Ag_a}^{Ag_b}$, the trustor Ag_a would update the given trust rates recorded in the vector $\underline{\mathcal{T}r}_{Ag_a}^{Ag_b}$ of the consulting agents $\mathcal{C}s_{Ag_a}^{Ag_b}$ to the rates recorded in the vector ($\underline{\mathcal{UPT}r}_{Ag_a}^{Ag_b}$) that if were considered in the trust assessment of the trustee Ag_b, would have leaded to the least possible difference between the estimated trust ($Tr_{Ag_a}^{Ag_b}$) and the observed trust ($OTr_{Ag_a}^{Ag_b}$). This can be formulated as an optimization problem that Ag_a resolves in order to update the given trust rates of consulting agents.

$$min_{\underline{\mathcal{UPT}r}_{Ag_a}^{Ag_b}} |Tr_{Ag_a}^{Ag_b} - OTr_{Ag_a}^{Ag_b}| \quad (7)$$

$$\text{subject to} \quad \mathcal{C}ons_{Ag_a}^{Ag_b}$$

We have to mention that resolving this optimization problem, results in building the vector $\underline{\mathcal{UPT}r}_{Ag_a}^{Ag_b}$, which means that we are not changing the present given rates $\underline{\mathcal{T}r}_{Ag_a}^{Ag_b}$, but just we keep the updated ones in the resulting vector.

3.2 Maintenance Process

In general, consulting agents may unintentionally provide accurate or inaccurate information. Therefore, it is not wise that Ag_a adjusts his belief set only considering one interaction (replacing the given rates with the updated ones). In this respect, Ag_a performs a periodic maintenance process to analyze the overall performance of the consulting agents that were involved in one or few trust assessment processes and thus there are as many updated trust ratings.

Basically, in the maintenance process, Ag_a has a vector ($\mathcal{I}tr_{Ag_a}^{Ag_i}$) for each particular consulting agent Ag_i, where each element of this vector contains the following information: (`Inf1`) a trustee agent (let us say Ag_b) that Ag_i provided a trust rate for him; (`Inf2`) this trust rate ($Tr_{Ag_i}^{Ag_b}$); (`Inf3`) the number of interactions between Ag_i and Ag_b ($NI_{Ag_i}^{Ag_b}$); and (`Inf4`) the updated trust rate Ag_a obtains by resolving the optimization problem discussed in previous subsection ($UP\mathcal{T}r_{Ag_a}^{Ag_i}$). We refer to each item of the vector element by $\mathcal{I}tr_{Ag_a}^{Ag_i}$.`Infj` ($1 \leq j \leq 4$).

Furthermore, Ag_a considers all the pairs of the given rates $Tr_{Ag_a}^{Ag_i}$ and his corresponding updated trust rate $UP\mathcal{T}r_{Ag_a}^{Ag_i}$. Figure 2 represents the maintenance algorithm done by Ag_a regarding to a particular consulting agent Ag_i. In the maintenance process, Ag_a saves all the updated trust ratings ($UP\mathcal{T}r_{Ag_a}^{Ag_i}$) obtained for Ag_i in the set $UP_{Ag_a}^{Ag_i}$. The values saved in the set are different updated values corresponding to each trust assessment process of Ag_a that involves Ag_i. However, the present given trust rate to Ag_i is still unchanged ($Tr_{Ag_a}^{Ag_i}$). The trustor Ag_a in order to check the coherency of the updated ratings would compute the average updated trust rating $\overline{UP\mathcal{T}r}_{Ag_a}^{Ag_i}$ and the standard deviation of the recorded updated trust ratings $\sigma_{UP\mathcal{T}r_{Ag_a}^{Ag_i}}$. If $\sigma_{UP\mathcal{T}r_{Ag_a}^{Ag_i}}$ is within the acceptable coherency error threshold φ, this means that the updated trust ratings reflect a trust rate that would make more sense for such a consulting agent. Therefore, Ag_a would replace his given trust rate to Ag_i by the average updated trust ratings $\overline{UP\mathcal{T}r}_{Ag_a}^{Ag_i}$ obtained from a number of trust assessment adjustment procedures. In contrast, if $\sigma_{UP\mathcal{T}r_{Ag_a}^{Ag_i}}$ is outbound the coherency error threshold φ, this means that the updated trust ratings are in diverse directions, which reflect the inconsistency of Ag_i in providing information. In this case, Ag_a would decrease his given rate to Ag_i by a ratio obtained from the portion of the average number of interactions done between the trustee Ag_b, the consulting agent Ag_i and the trustor Ag_a. The higher number of interactions these agents have, the more accurate information is supposed to be provided. As a result, Ag_a would decrease more the trust rate of these consulting agents if they provide inconsistent information.

It is important to discuss the importance of the maintenance process in the sense that we elaborate how the retrospect trust adjustment process could address the system inconsistency, and consequently how lack of such mechanism would face unavoidable degradation in the system efficiency. The proposed trust model is based on the combination of the suggestions about the credibility of the particular trustee agent. Being accurate, any time a trustor seeks the best combination method, which can possibly lead to the least estimation error. Performing the maintenance process, the trustor agent increases or decreases his trust rate about any consulting agent in the sense that the adjustment reflects the consulting agents' accuracies. Although the adjustment could overestimate (or underestimate) a particular consulting agent Ag_i, the trustor Ag_a would give the benefit of the doubt that Ag_i functions better (or worse) in consultation about the credibilities of some other agents. To this end, in spite of rating any interacting agent and then updating

---

function maintenance(Ag_a, Ag_i)
  for all *element X of the vector* $\mathcal{I}tr_{Ag_a}^{Ag_i}$
    $Ag_b = X.Inf1; \quad k = 0; \quad UP_{Ag_a}^{Ag_i} = \{\};$
    $TotalNI_{Ag_i} = 0;$
    $UR_k = X.Inf4;$
    $k++;$
    $UP_{Ag_a}^{Ag_i} = UP_{Ag_a}^{Ag_i} \bigcup \{UR_k\};$
    $TotalNI_{Ag_i} = TotalNI_{Ag_i} + NI_{Ag_i}^{Ag_b};$
    $TotalD_{Ag_i} = TotalD_{Ag_i} + |Tr_{Ag_i}^{Ag_b} - OTr_{Ag_a}^{Ag_b}|;$
  $\overline{UPTr}_{Ag_a}^{Ag_i} = \frac{\sum_{k=1}^{|UP_{Ag_a}^{Ag_i}|} UR_k}{|UP_{Ag_a}^{Ag_i}|};$
  $\sigma_{UPTr_{Ag_a}^{Ag_i}} = \frac{1}{|UP_{Ag_a}^{Ag_i}|}\sqrt{\sum_{k=1}^{|UP_{Ag_a}^{Ag_i}|}(UR_k - Tr_{Ag_a}^{Ag_i})^2};$
  if ($\sigma_{UPTr_{Ag_a}^{Ag_i}} < \varphi$)
    $Tr_{Ag_a}^{Ag_i} = \overline{UPTr}_{Ag_a}^{Ag_i};$
  else
    $\overline{NI}_{Ag_i} = \frac{TotalNI_{Ag_i}}{|UP_{Ag_a}^{Ag_i}|};$
    $\overline{D}_{Ag_i} = \frac{TotalD_{Ag_i}}{|UP_{Ag_a}^{Ag_i}|};$
    $Tr_{Ag_a}^{Ag_i} = Tr_{Ag_a}^{Ag_i} \times \frac{\overline{NI}_{Ag_i}}{\overline{NI}_{Ag_i} + NI_{Ag_a}^{Ag_i}} \times (1 - \overline{D}_{Ag_i});$

Figure 2: Maintenance algorithm for adjusting trust value of Ag_i by agent Ag_a.

the belief in a regular manner, Ag_a attracts the consulting agents that could possibly benefit in the overcoming trust estimation processes. Meanwhile, Ag_a discards the ones that could possibly distract the overcoming processes.

Let us imagine a mechanism without maintenance. In such a model, the trust propagation would be the solution for evaluating the credibility of a particular trustee agent Ag_b. Suppose Ag_b has already been evaluated and is interacting with others by providing high quality services. For some reasons, this agent changes his intentions and does not provide such quality services anymore. Therefore, a trustor agent that obtains the bad service starts to rate bad for such trustee agent. These ratings would be accumulated with previous ratings (clearly good) in the belief set of the trustor agent. Therefore, it would take some certain number of interactions that the trustor agent updates his current belief about Ag_b ($Tr1_{Ag_a}^{Ag_b}$) to a new trust rate ($Tr2_{Ag_a}^{Ag_b}$). For this adjustment, Ag_a needs to accumulate extra bad ratings about Ag_b, which would cause $Tr2_{Ag_a}^{Ag_b}$ to be less than $Tr1_{Ag_a}^{Ag_b}$. We declare the extra bad ratings by $b'^{Ag_b}_{Ag_a}$ and compute it in equation 8. In this equation $g_{Tr1_{Ag_a}^{Ag_b}}$ represents number of good ratings of the first trust value. Likewise, if a trustee agent Ag_b changes his quality of service from bad to good, then a certain number of extra good ratings $g'^{Ag_b}_{Ag_a}$ are to be accumulated in order to increase the current rate. We compute the extra number of good ratings to enhance $Tr1_{Ag_a}^{Ag_b}$ to $Tr2_{Ag_a}^{Ag_b}$ in equation 9.

$$b'^{Ag_b}_{Ag_a} = \frac{Tr1_{Ag_a}^{Ag_b} - Tr2_{Ag_a}^{Ag_b}}{Tr1_{Ag_a}^{Ag_b} \times Tr2_{Ag_a}^{Ag_b}} g_{Tr1_{Ag_a}^{Ag_b}} \tag{8}$$

$$g'^{Ag_b}_{Ag_a} = \frac{Tr1_{Ag_a}^{Ag_b} - Tr2_{Ag_a}^{Ag_b}}{Tr1_{Ag_a}^{Ag_b} \times Tr2_{Ag_a}^{Ag_b} - Tr1_{Ag_a}^{Ag_b}} g_{Tr1_{Ag_a}^{Ag_b}} \tag{9}$$

Table 1: Simulation summarization over the obtained measurements.

	S.P. Agent Type	Density in S.P. Community	Provided Utility	
			Range	Standard Deviation
Service Provider Agents (S.P.)	Good	15.0%	]+5, +10]	1.0
	Ordinary	30.0%	]-5, +5]	2.0
	Bad	15.0%	]-10, -5]	2.0
	Fickle	40.0%	[-10, +10]	-
Service Consumer Agents (S.C.)	S.C. Agent Type	Density in S.C. Community	Number of Joining and Leaving Agents at Each RUN	
	Proposed Model	33.3%	10 (5.0%)	
	Travos	33.3%	10 (5.0%)	
	BRS	33.3%	10 (5.0%)	

Here the trustor Ag_a would need to perform at least the $\lceil b'^{Ag_b}_{Ag_a} \rceil$ (or $\lceil g'^{Ag_b}_{Ag_a} \rceil$) number of interactions to change his belief from $Tr1^{Ag_b}_{Ag_a}$ to $Tr2^{Ag_b}_{Ag_a}$ and then upon propagation, he distributes his new belief about Ag_b. This basically shows the weakness of such rating mechanisms in the environment with high rate of dynamism (rate of change is higher than rate of adaptation). In this case, the agents are unsure about their beliefs as long as they do not reach their belief stability.

4. EXPERIMENTAL RESULTS

4.1 The Testbed and Experimental Results

In this section, we describe the implementation of proof of concept prototype. In the implemented prototype, agents are implemented as $Jadex^{©TM}$ agents, i.e. they inherit from the basic class called $Jadex - Simulator^{©TM}$ $Agent$. The agent reasoning capabilities are implemented as Java modules using logic programming techniques. Like in [5], the testbed environment (represented in table 1) is populated with two agent types: (1) *service provider agents* that are supposed to provide services (toward simplicity, we assume only one type of service is provided and therefore consumed); and (2) *service consumer agents* (equipped with different trust models) that are seeking the service providers to interact with and consume the provided service. The simulation consists of a number of consequent RUNs in which agents are activated and build their private knowledge, keep interacting with one another, and enhance their overall knowledge about the environment. Table 1 represents four types of the service providers we consider in our simulation: good, ordinary, bad and fickle. The first three provide the service regarding to the assigned mean value of quality with a small range of deviation. Fickle providers are more flexible as their range of service quality covers the whole possible outcomes. Upon interaction with service providers, service consumer agents obtain a utility (called *gained utility*).

After each interaction, the service consumer agent rates the service provider by evaluating the provided service quality. The rating is sent to the provider as feedback and accumulated in the service consumer's belief set. The accumulated ratings enable the consumer to reason about the upcoming service provider selection. Also he is able to provide personal rating about a particular provider if he has been asked by another agent to provide his suggestion. In the simulation environment, agents are equipped with different trust models in the sense that their provider selection policies are different. In our experiment, we compare the effectiveness of the proposed model agents with agents that are equipped with other trust models in different perspectives, in the sense that their overall performance comparison could be obtained.

4.2 Overall Performance Comparison

In order to discuss the proposed model's overall performance, we compare it with BRS [1] [11] and Travos [2] [12] trust models. We provide a detailed performance discussion of these trust models in Section 5. To express the proposed model properties in a more clear way, we use high number of fickle agents, making a biased environment. Doing so, we compare the trust models concerning how they survive in such an environment, where agents constantly change their behaviors. Travos and BRS are similar to the proposed model in the sense that they do consider other agents' suggestions while evaluating the trust of some specific agent and discard inaccurate suggestions aiming to adapt themselves to the environment inconsistency attitude. However, Travos and BRS models differ from ours in the trust assessment mechanism and analysis they perform in order to choose the best possible provider. At the end, the utility gained by each model is considered as its efficiency in selecting reliable service providers. The experimental measurements of the comparison between these models are outlined in table 2 and a graph representing the cumulative utility gained of the three models is illustrated in Figure 3-a. The experimental results show that the proposed model agents outperform others in selecting best providers and thus gaining more utility. This can be explained by the fact that in such a biased environment, finding the best provider is a challenging issue. The proposed model agents are equipped with a mechanism that enables them to adapt with the environment faster than regular rating mechanism and its distribution. We will discuss the effectiveness of the proposed model in more details in the following sections.

4.3 Proposed Model Performance

In the proposed model, we try to establish a trust mechanism where an agent, firstly can maintain an effective trust assessment process and secondly, accurately updates his belief set, which reflects the other agents likely accuracy. In order to confirm the mentioned characteristics, we compare the proposed model with Travos and BRS trust models in two perspectives. In former comparison view, we use the agents that only perform a comprehensive trust assessment process. We refer to this group of agents as *Comprehensive Trust Group* (CTG). In later overview, we use the agents that are (in addition to the comprehensive trust assessment mechanism), capable of performing the periodic maintenance in order to increase their adaptivity. We refer to this group of agents as *Maintenance Trust Group* (MTG).

First we compare the models in terms of good provider se-

[1]BRS trust model collects the after-interaction ratings and estimates the trust using beta distribution method. This trust model ignores the ratings from such agents that deviate the most from the majority of the ratings.

[2]Travos trust model is similar to BRS in collecting the after-interaction ratings and estimating the trust using beta distribution method. But Travos ignores the ratings from agents that provide intermittent reports in the form of suggestions.

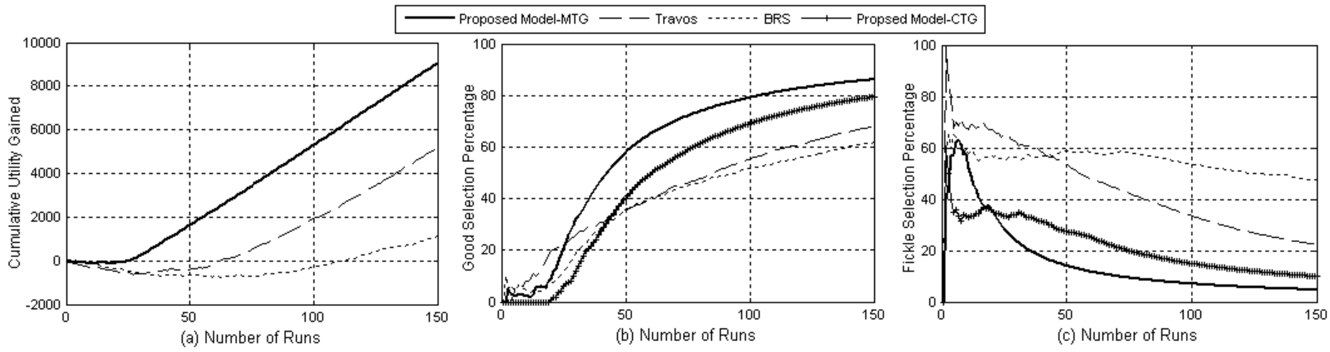

Figure 3: Overall comparison of the proposed model with Travos and BRS in terms of (a) cumulative utility gained; (b) good selection percentage; and (c) fickle selection percentage.

Table 2: Results summarization over the obtained measurements.

Measurements and Characteristics	Proposed Model	Travos	BRS
No. of active agents in simulation	20	20	20
No. of RUNs in each simulation	300	300	300
Measured cumulative utility gained in five simulations	9,648	6,032	2,870
	9,721	6,736	1,678
	9,939	7,455	2,188
	9,652	5,909	1,573
	9,388	7,735	1,760
Average cumulative utility gained	*9,669.6*	*6,773.4*	*2,013.8*
Standard deviation of cumulative utility gained	176.23	733.14	476.42
Half value of confidence interval	218.52	909.09	590.76
Full interval with 95% confidence level	**[9,451 - 9,888]**	**[5,864 - 7,682]**	**[1,423 - 2,604]**

lection percentage. In such a biased environment, the number of good providers are comparatively low. Therefore, the agents need to perform an accurate trust assessment to recognize the best providers. As it is clear from the Figures 3-b and 3-c, CTG agents function better than Travos and BRS. The reason is that in this model, agents are assessing the credibility of the providers using other agents suggestions depending on their credibility and to what extent they know the provider. Afterwards these agents rate the provider, which would be distributed to other agents upon their request. Not excluding the fact that CTG agents are considering partial ratings for consulting agents, we state that they weakly function when the environment contains agents that constantly change their intentions. Therefore, the previous history would not reflect the likely credibility of such agents in the future.

MTG agents in addition to the comprehensive trust assessment, provide a periodic maintenance process, which enables them to effectively sense the environment changes and thus adapt themselves faster than other models. Figure 3-b shows that MTG agents outperform other models in best provider selection. Relatively, Figure 3-c shows MTG agents' least selection of fickle providers. This is expressed by the fact that MTG agents recognize the providers that recently have changed their service qualities.

We illustrate this feature in Figure 4, which depicts the percentage of selecting some providers (that are dynamically changing their behaviors) by MTG agents vs. elapsing RUNs in the simulation. In this graph, two good and two fickle providers ($Pr.g1$, $Pr.g2$, $Pr.f1$ and $Pr.f2$) are considered to change their behaviors. The two good (resp. the two fickle) agents are similar except in the rate of their behavior change (0.3 vs. 0.45). The adaptivity of MTG agents is observable in the sense that after certain number of RUNs, they adapt themselves with the new quality of service provided. For example at point P_1 (resp. P_2), $Pr.g1$ (resp. $Pr.g2$) starts to change his behavior. We observe that after this point, the selection percentage of this agent drops, which reflects the property that MTG agents start to adapt themselves with the new behavior of this agent. The same adaptation is observed for $Pr.f1$ after p_3 and $Pr.f2$ after p_4. In general, MTG agents discard good providers when they start providing low quality and in contrast start reselecting fickle providers when they provide high quality of service. Obviously, because of intermittent attitude of the fickle providers, MTG agents would consider longer time to completely count on them. Therefore, their selection percentage is less than good providers. In this case, such a fickle provider will not reach %100 selection percentage (p_5) because his rate of behavior change is higher than his stability rate for MTG agents.

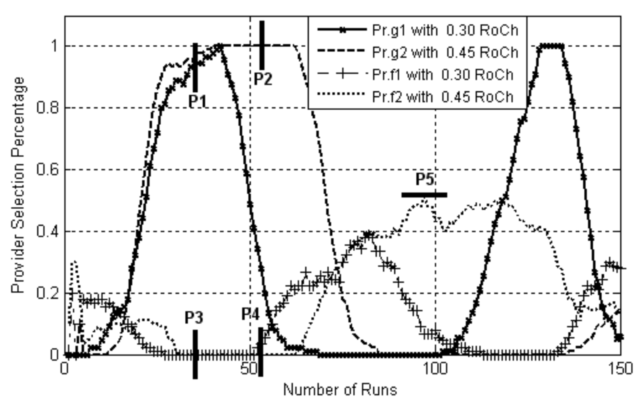

Figure 4: Good and Fickle provider selection percentage in two *R*ates *o*f behavior *Ch*ange ($RoCh : 0.3$ and $RoCh : 0.45$).

5. RELATED WORK

Perhaps the best-known approaches to trust using witness's ideas in multi-agent systems are FIRE [5], SPORAS [9], Referral [7], Regret [8], Beta Reputation System [11] , and TRAVOS [12]. Generally speaking, agents are aimed to make estimation and prediction independently. The issue is that there are always fickle agents that try to distract the overall process. These agents can either try to slander other good agents by lying about their trust levels or supporting bad agents by exaggerating about their credibility.

In BRS model, the trustor agent in the trust assessment process uses beta distribution method and discards the ratings that deviate the most from the majority of the ratings. Concerning this, BRS is comparatively a static trust method, which causes a low-efficient performance in very dynamic (biased) environment. In general, this model is not sensitive to an agile behavior change. This means that if a BRS agent decides to evaluate an agent that he is not acquainted with, he considers the majority of ratings, which are supposed to be truthfully revealed about the trustee agent. In such a case that the trustee agent has just changed his strategy, the trustor agent would loose in trust assessment and does not maintain any action to verify the accuracy of the gained information. It may take as much time that other agents perform a number of direct interactions to start rating about the spurious trustee agent. Therefore, as illustrated in figure 3-b, the BRS agents would have less percentage of good providers selection and relatively higher percentage of fickle providers selection (illustrated in figure 3-c). Generally, it would take more time for BRS agents to adapt themselves to the new environment conditions.

Travos [12] trust model is similar to BRS in using beta distribution to estimate the trust based on the previous interactions. Travos model also does not have partial rating. Hence, the trustor agent merges his own experience with recommendations from other agents. However, unlike BRS model, Travos filters the surrounding agents that are fluctuating in their reports about a specific trustee agent. To some extent, this feature would cause a partial suggestion consideration and thus, Travos agents would adapt faster comparing to BRS agents. Rates concerning the good and fickle selection percentage shown in figures 3-b and 3-c reflect higher efficiency of Travos compared to BRS. However, Travos model considers that agents do not change their behavior towards the elapsing time. These missing assumptions affect the accuracy of trust estimation in a very biased environment. This is the case when a surrounding agent is being discarded because of providing diverse reports about a particular trustee agent. In this case, the deviation would be filtered by mistake if the reports are reflecting the fickle attitude of that particular provider.

6. CONCLUSION

The contribution of this paper is the proposition of a new probabilistic-based trust model to secure multi-agent systems. The trust assessment procedure is composed of comprehensive trust evaluation and retrospect adjustment. Comprehensive approach is based on integrating suggestion of consulting agents, objectively enhancing the accuracy of agents to make use of the information communicated to them. Retrospect process considers the communicated information to judge the accuracy of the consulting agents in the previous comprehensive trust evaluation process.

Our model has the advantage of being computationally efficient as it takes into account the important factors involved in the trust assessment process. Moreover, extra process of maintenance enables agents to dynamically adjust their belief, and consequently update their trustworthy community in a more efficient manner. The proposed mechanism is compared with other related models and discussed in details to prove its capabilities and efficiency. Our plan for future work is to advance the assessment model to enhance this efficiency using argumentation techniques [3, 13]. In the retrospect process we need to elaborate more on the optimization part, trying to formulate it in the sense to be adaptable to diverse situations. We plan to consider also the dynamic change of agents' behaviors. We need to analyze in depth the affect of diverse strategies in selections. Finally, we plan to maintain more detailed analysis in comparison with other models to capture more results reflecting the proposed model capabilities.

7. REFERENCES

[1] J. Bentahar, B. Khosravifar. Using Trustworthy and Referee Agents to Secure Multi-Agent Systems. In Proceedings of the 5'th International Conference on Information Technology: New Generations, IEEE pp. 477-482, USA, 2008.

[2] D. Marchiori and M. Warglien. Predicting Human Interactive Learning by Regret-Driven Neural Networks. Science, 319(5866):1111-1113, 2008.

[3] P.M. Dung, P. Mancarella and F. Toni. Computing ideal sceptical argumentation. Artificial Intelligence, Special Issue on Argumentation in Artificial Intelligence, 171(10-15):642-674, 2007.

[4] T. Dong-Huynh, N.R. Jennings and N.R. Shadbolt. Certified reputation: How an agent can trust a stranger. In Proceedings of The 5'th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1217-1224, Japan, 2006.

[5] T. Dong-Huynh, N.R. Jennings and N.R. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. Journal of Autonomous Agents and Multi-Agent Systems 13(2):119-154, 2006.

[6] Y. Wang, and M.P. Singh. Formal trust model for multiagent ststems. Proceedings of the 20'th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1551-1556, 2007.

[7] P. Yolum and M.P. Singh. Engineering self-organizing referral networks for trustworthy service selection. IEEE Transaction on Systems, Man, and Cybernetics, 35(3):396-407, 2005.

[8] J. Sabatar. Trust and reputation for agent societies. Phd thesis, Universitat autonoma de Barcelona, 2003.

[9] G. Zacharia, and P. Maes. Trust management through reputation mechanisms. Applied artifitial intelligence, 14(9):881-908, 2000.

[10] J. Sabater, M. Paolucci and R. Conte. Repage: REPutation and ImAGE Among Limited Autonomous Partners. Journal of Artificial Societies and Social Simulation 9(2), 2006.

[11] A. Jesang and R. Ismail. The beta reputation system. 15'th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, 2002.

[12] W. T. Teacy, J. Patel, N.R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agentsand Multi-Agent Systems, 12(2):183-198, 2006.

[13] J. Bentahar, Z. Maamar, D. Benslimane, and Ph. Thiran. An Argumentation Framework for Communities of Web Services. In IEEE Intelligent Systems, 22(6): 75-83, 2007.