An Approach to Comprehensive Trust Management in Multi-Agent Systems with Credibility

Babak Khosravifar, Jamal Bentahar, Maziar Gomrokchi and Rafy Alam
Concordia Institute for Information Systems Engineering, Concordia University, Canada
b_khosr@encs.concordia.ca, bentahar@ciise.concordia.ca, m_gomrok@encs.concordia.ca, al_muha@encs.concordia.ca

***Abstract*—Security is a substantial concept in multi-agent systems where agents dynamically enter and leave the system. Different models of trust have been proposed to assist agents in deciding whether to interact with requesters who are not known (or not very well known) by the service provider. To this end, in this paper we progress our work on security for agent-based systems, which is embedded in service provider's trust evaluation of the counter part. Agents are autonomous software equipped with advanced communication (using public dialogue game-based protocols and private strategies on how to use these protocols) and reasoning capabilities. The service provider agent obtains reports provided by *trustworthy agents* (regarding to direct interaction histories) and *referee agents* (in the form of recommendations) and combines a number of measurements, such as number of interactions and timely relevance, to provide an overall estimation of a particular agent's likely behavior. Requesting this agent, called the *target agent*, to provide the number of interactions it had with each agent, the service provider penalizes the agents who lied about having information for trust evaluation process. In addition, after a periodic time, the actual behavior of the target agent is compared against the information provided by others. This comparison leads to both adjusting the credibility of the contributing agents in trust evaluation and improving the system trust evaluation by minimizing the estimation error. Overall the proposed framework is shown to assist agents effectively perform the trust estimation of interacting agents.**

***Index Terms*—Trust, Multi-Agent Systems, Agent Communication, Dialogue Games.**

I. Introduction

During the past couple of years, agent communication languages and protocols have been of much interest in multi-agent systems. Agents are distributed in large scale network and mutually interact to collaborate, coordinate and share services with other agents. Therefore trust is essential in effective interactions within open multi-agent systems [5], [17], [15]. In other words, an agents trust in another is the measure of willingness that the agent will make what it agrees to do. Generally in trust-based approaches the central control unit is avoided to optimize the efficiency of the overall system. Therefore a rational agent maintain autonomous operations with the contributing agents which would be substantially the base of their rely in one another. Obviously the mutual trust is subject to change regarding to time and frequent interactions taken place.

To maintain a trust-based approach, we propose a framework allowing agents to represent the trust they have in one another. This paper is the continuation of the previous work which was taking a narrower view of trust, representing a set of trust meta-data to define the trust level of the contributing agents [4], [5]. To do so, agents mutually interact and rate each other based on the interaction done (either satisfactory or dissatisfactory). The obtained ratings are accumulated to make the trustworthiness of a particular agent. Inter-agent communication is regulated by protocols (shared amongst agents and thus public) and determined by strategies (internal to agents and thus private). Using this framework, agents are capable of evaluating the trust level of the agents which are not known (or not very well known) by consulting other agents who can provide suggestions about the trustworthiness level of other agents. The idea of consulting with others originates from the fact that agents by nature assess diverse trust levels of an agent depending on their different experiments of direct interaction with that specific agent. Therefore the trust concept in conventional mechanism-design approaches such as Groves and Vickery-Clarke-Groves (VCG) mechanisms [18] would fail as they hold that agents refer to their public and inter-dependent information to define the trust estimation regarding to other agents.

Centralization is the process by which the decision makings become centralized in a particular location, whereas decentralization is the process of distributing decision makings amongst the system components. In multi-agent systems also decentralized trust models have been always preferred as centralized approaches fail to adequately address the e-computing challenges posed by open systems. However decentralized trust models are purely qualitative and consider agents as objects interacting by message exchange, without reasoning capabilities. Generally in multi-agent systems agents reason using their current knowledge bases (private and independent information) before making decisions, and can thus engage in flexible interactions [5]. In addition some of these existing models do not consider limitations in terms of false information provided or lack of information to perform evaluation process. They also do not provide trust propagation through the system.

Generally trust models using direct experience need long term of interaction to reach a state that agents can evaluate trust level of each other. Moreover trust models using witness reports usually do not take care of the risks that the witnesses collude with the *target agents* (i.e. the agents to be evaluated) and provide fake information to support them. Therefore it is

important to be able to evaluate the credibility of the witnesses [4], [8], [11]. This is done either by direct experience used to estimate the trust level of these agents or by moving to the second level of evaluation process, asking other agents that are known to be trustworthy about the credibility of the witnesses. However, there is a problem if such *trustworthy agents* are not able to report on the testimony agents. Moreover the trust is not a transitive relationship (when agent A is trustworthy according to agent B and agent B is trustworthy according to agent C, does not mean that agent A is also trustworthy according to agent C). This paper aims at overcoming these limitations by proposing a framework combining the use of *trustworthy agents* and *referee agents* proposed by the target agent. Also using the framework, the *requesting agents* (i.e. the agents requesting information about a target agent) perform maintenance after a period of direct interaction with a new agent in order to adjust the trustworthiness of the consulting agents who provided information regarding to the trust level of the new agent. In the maintenance process, the suggestions provided by other agents are compared with the actual behavior of the new agent in direct interaction.

The remainder of this paper is organized as follows. In Section II, we present the mathematical formalization of the problem, in particular *the trust function* and the notion of *direct trust*. Section III focuses on the propagation of trust through a social network and defines our framework that combines trustworthy and referee agents as reporters. In Section IV, we describe and discuss the details of computing the trust in the combined framework. Section V shows some properties of our model from a probabilistic point of view and highlights the situations of more accurate trust estimation of the testimony agent. Section VI discusses the concept of penalizing the agents who lie about having information regarding the target's trust. In Section VII, we perform the maintenance the service provider makes after a certain amount of time after the interactions initiated. Section VIII briefly discusses the proof of concepts prototype. Section IX compares our framework to related work and Section X concludes the paper.

II. Mathematical Formalization

In our framework, agents are equipped with Beliefs, Desires and Intentions (BDI). They use the BDI architecture when they interact with each other. They are also equipped with reasoning capabilities allowing them to evaluate their interactions and to decide about the communicative acts to perform during interactions. These reasoning capabilities could be implemented using different theories such as argumentation [6], [7], game theory, decision theory [14], etc. The purpose of this paper is not to elaborate on these capabilities, but we simply assume their existence. To interact, agents use dialogue games, which are logical rules specifying the communication protocol.

During the past couple of years, the trust concept has been of more attention in the multi-agent systems where entities are autonomous and heterogenous. In this paper we adopt a probabilistic-based approach to compute trust values [5] before and after agent interaction. We define an agent's trust in other agents as a probability function as follows:

Definition 1: Let $\mathcal{A}$ be a set of agents, and $\mathcal{D}$ be a set of domains or topics. The trust function Tr associates two agents from $\mathcal{A}$ and a domain from $\mathcal{D}$ with a trust value between 0 and 1:

$$Tr : \mathcal{A} \times \mathcal{A} \times \mathcal{D} \longrightarrow [0,1]$$

Given some concrete agents Ag_a and Ag_b in $\mathcal{A}$ and some concrete domain D, $Tr(Ag_a, Ag_b, D)$ stands for "the trust value associated to the target agent Ag_b in domain D by the requesting agent Ag_a".

The scale and dynamism of open environment make the participants able to rank each other's reputation level as they keep up interacting. This is done by mutual rating (+1,-1) of both contributing parties over the previous interaction. Therefore agents rank each other +1 if they were satisfied of the other's provided service (this could be in terms of price, delivery in time and condition); respectively if a conflict happens between two agents, this will affect the confidence they have about each other by rating -1. However, this is only related to the domain $\mathcal{D}$, and not generalized to other domains in which the two agents can trust each other. It is obvious that judging based on the accumulated ratings would represent unfairness, as all the interactions would be treated equally and factors like time and the value of the transaction are not considered. Therefore some trust metrics are to be taken into account to adjust the confidence to some certain extent.

To simplify the notation, in the remainder we will omit the domain from all the formulas. Given agents Ag_a and Ag_b in $\mathcal{A}$, we will represent $Tr(Ag_a, Ag_b)$ in short as $Tr_{Ag_a}^{Ag_b}$.

In this section we consider the case where agents in the system know each other because they had a prior interaction history and can thus compute the trust value of all agents (and thus the Tr function) *directly*. Using their reasoning capabilities, agents evaluate the outcomes of their interactions. Let us assume that they can evaluate their interactions as "*good*" or "*bad*". A good interaction could be one after which the agent is satisfied because his goal that prompted the interaction is achieved after the interaction (successful outcome). A bad interaction could be the opposite (unsuccessful outcome).

In general, agents can evaluate the outcomes of their interactions using more flexible values such as "*very good*", "*good*", "*fair*", "*bad*", and "*very bad*". This would generates real numbers which fall in the range [0,1] and thus instead of just binary rating (-1 or +1) we would have more flexible real ratings which represent the satisfactoriness or not satisfactoriness of the outcome. In the general case, agent can evaluate their interactions 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 successful (for example of type "*very good*", "*good*", and "*fair*"). Let $NI_i^{Ag_b}_{Ag_a}$ be the number of interactions of type i that Ag_a had with Ag_b. Then Tr can be computed by Equation 1 below as the ratio of the "*number of successful outcomes*" to the "*total number of possible outcomes*":

$$Tr_{Ag_a}^{Ag_b} = \frac{\sum_{i=1}^{m} w_i v_i NI_i{}_{Ag_a}^{Ag_b}}{\sum_{i=1}^{n} w_i v_i NI_i{}_{Ag_a}^{Ag_b}} \quad (1)$$

where w_i is the weight associated to the interaction type i and the v_i is the measure reflecting the importance of the interaction. Agents can use several strategies when weighting the interaction types. For example, to minimize the risk of dealing with *untrustworthy agents*, the weight w of "*very bad*" interactions could be higher than the one of "*very good*" interactions. Therefore unsuccessful interactions are more valuable when assessing the agents' trust, and agents should perform well and avoid bad behavior in order to get a better trust value. However, less demanding agents could give the same weight w to all interaction types, or give more weight to the "*very good*" and "*good*" interactions. Regarding the variable v_i, in the case of interactions about some transactions, this variable is used to avoid two transactions with different values being treated equally. The important transactions are associated to the higher values. This idea will protect the model from attacks like reputation squeeze [27] in which one agent would obtain some positive ratings and make a bad interaction which actually makes a large damage. v_i can be defined in different ways, for example zhang [19] evaluates the transaction weight by $\nu[p(x,y)] = p(x,y)/\mu$ in which the $p(x,y)$ is the transaction value between agents x and y and μ denotes the standard price that the system assigns to Escrow service (for example in eBay, μ = \$200). Consequently, a transaction with value \$5000 is more important than a transaction with value \$250. Ag_a will consider Ag_b as trustworthy if $Tr_{Ag_a}^{Ag_b}$ is higher or equal to a *trustworthiness threshold* fixed by Ag_a. The assigned threshold depends on the objective of Ag_a to restrict his network of trustworthy agents (by using high threshold) or to have a large network (by using low threshold).

In open multi-agent systems, agents are known to be autonomous and may not always complete tasks that are requested from them. Normally they do not know everything about their dynamic environment and there is no central manager to control all the agents. Therefore trust is set between two agents which are supposed to interact. Trustworthiness is a dynamic characteristic that changes according to the interactions taking place between two agents Ag_a and Ag_b. Agents can evaluate directly the trust value of agents they have interacted with extensively. This evaluation denotes the agents overall idea about the service provided by the other party in terms of cost or payment, availability, service condition, delivery, etc. However, if the number of interactions with some agent is low (e.g. because the agent has only recently joined the system), agents are not able to compute their trust value directly, but may need to rely upon information provided to them by other agents (that may have interacted more extensively with the given agent). Different protocols have been emerged capable of consulting other agents in order to get a better idea about a particular agent's trust level. As proposed in [1], [3], [8], each agent has two kinds of beliefs when evaluating the trustworthiness of another agent: local beliefs and total beliefs. Local beliefs are based on the direct experience of interaction agents. Total beliefs are based on the combination of the different testimonies of other agents that we call *witnesses*. In our framework, local beliefs are given by Equation 1. Total beliefs require studying how different probability measures offered by witnesses can be combined. We use two kinds of witnesses: *trustworthy agents* that the requesting agent trusts, and *referee agents* that the target agent introduces to report on his trust. Our framework adjusts the trust level of the unknown or not well known agent based on the type of the consulting agents as they are directly known or trusted by the requesting agents or not. The suggestions gathered from different types of consulting agents are counted based on the credibility of the agent providing the information. In this model we consider the following metrics which affect the information provided for the trust evaluation process: the timely relevance of information transmitted by contributing agents and the relationship quality denoted as the number of interactions done between the contributing and the target agents.

III. Trust Model

Suppose that an agent Ag_a wants to evaluate the trustworthiness of a target agent Ag_b with who he never (or not enough) interacted before. Ag_a may want to consult some other agents to get better and more accurate information about Ag_b's reputation. As illustrated in figure 1, this agent (*the requesting agent*) must ask some other agents as third parties to provide information which would lead to estimate a more accurate trustworthiness level of Ag_b. The interfering agents are either known by Ag_a to be trustworthy (we call these agents *trustworthy agents*) or known by Ag_b and have been introduced by him to report on his trust level based on their past experience (we call these agents *referee agents*). Consequently, we distinguish the community of *trustworthy agents* from the community of *referee agents*. However there might be a *referee agent* which is also a trustworthy agent. This gives more chance to Ag_b as he introduced a more trustworthy and well-known agent as referee.

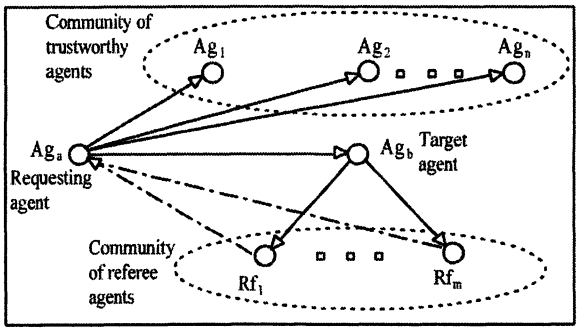

Fig. 1. Trustworthy and referee agents topology.

In this model, agents actively interact and find such referees in order to gain the trust of their potential partners. To do so they select the agents who has been given the best ratings to them. Ag_a respectively attributes a trustworthiness measure to each *trustworthy agent* Ag_i and *referee agent* Rf_j. When a *trustworthy agent* Ag_i is consulted by Ag_a, he provides a trustworthiness value for Ag_b if Ag_i knows Ag_b. *Referee agents* are more likely to recommend Ag_b as they have been introduced by him, but they should be known by Ag_a as well to be considered in the evaluation. Both *trustworthy agents*

and *referee agents* use their local beliefs to assess this value (Equation 1). Thus, the problem consists in evaluating Ag_b's trust level using the trustworthiness values transmitted by *trustworthy* and *referee agents*. Figure 2 illustrates the protocol agents use to gather information about the trust level of a target agent.

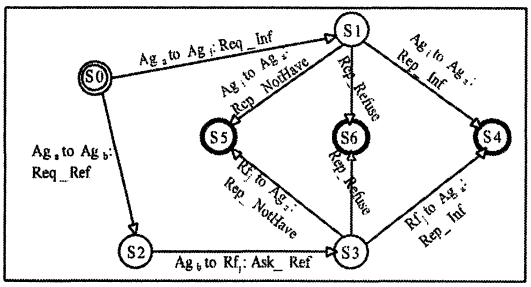

Fig. 2. Protocol of gathering information from trustworthy and referee agents

Ag_a uses the $Req\_Inf(Ag_a, Ag_i, Trust(Ag_b), t_0)$ communicative act which means Ag_a initially (at time t_0) sends to Ag_i (which is a *trustworthy agent*) a request for information related to Ag_b's trust. When Ag_i receives the Req_Inf communicative act, he uses the following rule, called Req_Inf dialogue game, to reply:

$$Req\_Inf(Ag_a, Ag_i, Trust(Ag_b), t) \Rightarrow$$
$$Rep\_Inf(Ag_i, Ag_a, Inf(Ag_b), t')$$
$$\vee\ Rep\_NotHave(Ag_i, Ag_a, *, t')$$
$$\vee\ Rep\_Refuse(Ag_i, Ag_a, *, t')$$

$Rep\_Inf(Ag_i, Ag_a, Inf(Ag_b), t')$ is the communicative act Ag_i uses to provide Ag_a with the relevant information regarding Ag_b's trust. $Rep\_NotHave(Ag_i, Ag_a, *, t')$ means that Ag_i does not have the relevant information about Ag_b's trust. $Rep\_Refuse(Ag_i, Ag_a, *, t')$ means that Ag_i refuses, for some reasons, to disclose the information he has about Ag_b's trust (the content of these two communicative acts is empty, represented by $*$). Here Ag_i may lie when he uses $Rep\_NotHave$ in order to affect Ag_b's trustworthiness. This issue originates the idea of asking Ag_b about his knowledge of Ag_i. Some penalties are applied to Ag_i when this untruthful behavior is discovered. This issue is discussed in more details in Section VI.

Meanwhile Ag_a uses the $Req\_Ref(Ag_a, Ag_b, t_0)$ communicative act which means Ag_a initially (at time t_0) sends to Ag_b a request for some *referee agents* who can recommend Ag_b. Ag_b is supposed to introduce some *referee agents* who support him in the trust evaluation done by Ag_a. Therefore Ag_b in order to choose its *referee agents* picks up the best ratings he had received during the past direct interaction experiences. Let $\mathcal{R}_{Ag_b}$ be the set of Ag_b's *referee agents*. Then Ag_b after receiving the Req_Ref communicative act, chooses the appropriate *referee agents* from $\mathcal{R}_{Ag_b}$. R_{Ag_b} denotes the set of these appropriate *referee agents*. Ag_a restricts $\mathcal{R}_{Ag_b}$ in terms of number of *referee agents* needed and best trustworthiness level. The following dialogue game specifies this issue:

$$Req\_Ref(Ag_a, Ag_b, t_0) \Rightarrow$$
$$\forall Rf_j \in R_{Ag_b}, Ask\_Ref(Ag_b, Rf_j, Ag_a, t')$$

$Ask\_Ref(Ag_b, Rf_j, Ag_a, t')$ is the communicative act to be used by Ag_b to ask Rf_j to recommend him to Ag_a with the relevant information regarding his trust. When Rf_j receives the Ask_Ref communicative act, he uses the following dialogue game to reply:

$$Ask\_Ref(Ag_b, Rf_j, Ag_a, t') \Rightarrow$$
$$Rep\_Inf(Rf_j, Ag_a, Inf(Ag_b), t'')$$
$$\vee\ Rep\_NotHave(Rf_j, Ag_a, *, t'')$$
$$\vee\ Rep\_Refuse(Rf_j, Ag_a, *, t'')$$

The *referee agent* will either provide the required information regarding to the trust level of Ag_b by using $Rep\_Inf(Rf_j, Ag_a, Inf(Ag_b), t'')$ communicative act or refuse to provide any information to Ag_a by using $Rep\_NotHave(Rf_j, Ag_a, *, t'')$ or $Rep\_Refuse(Rf_j, Ag_a, *, t'')$ communicative acts. It is rare that the *referee agent* do not have information regarding to the trust level of Ag_b because the agent has been chosen by Ag_b and seems there was a direct experience between the *referee agent* Rf_j and target agent Ag_b and Ag_b has got a good rating that he would think is supportive of his trust evaluation done by Ag_a. However, the *referee agent* can simply refuse disclosing this information for some private reasons.

Ag_a attributes a trust measure $Tr_{Ag_a}^{Ag_i}$ to each of the agents $Ag_i (i = 1 \ldots k)$ he considers trustworthy, and a trust measure $Tr_{Ag_a}^{Rf_j}$ to the *referee agents* $Rf_j (j = 1 \ldots k')$ he knows. In general, when an (*evaluator*) agent assesses the trustworthiness of another (*evaluated*) agent, the former may consider the latter either trustworthy or untrustworthy depending on the trust measure he assigns to this evaluated agent and some *threshold* fixed by the evaluator. The trust measure can be computed using Equation 1. We will define Ag_i (resp. Rf_j) trustworthy by Ag_a when the trust measure $Tr_{Ag_a}^{Ag_i}$ (resp. $Tr_{Ag_a}^{Rf_j}$), given by Equation 1, is greater than a threshold W_{Ag} (resp. W_{Rf}) fixed by Ag_a.

Sometimes Ag_b is not very well known and the *trustworthy agents* Ag_i also do not know Ag_b, therefore suggesting recommenders would be helpful in the Ag_b's evaluation process. As long as Ag_a requests its *trustworthy agents*, it asks Ag_b whether it can provide some referees. Once the *referee agents* directly introduced themselves to Ag_a, Ag_a will assess their trustworthiness likewise. There is a possibility that the *referee agent* is not that reliable since the *referee agent* can collude with agents that it intends to support by providing some falsely references for them. On the other hand, Ag_b chooses its *referee agents* to put forward and a rational agent only presents its best ones. Therefore Ag_a can expect some exaggerated information regarding to Ag_b's creditability. So far the *referee agents* provide a partial perspective on Ag_b's trust evaluation which would be quite useful in the absence of other resources.

We assume that consulting agents Ag_i and Rf_j also use equation 1 to assess the trust value of the agents they know,

and in particular Ag_b. Thus, the problem consists in Ag_a evaluating Ag_b's trust measure combining the trust values transmitted by *trustworthy* and *referee agents* to Ag_a. Next section elaborates on the computing methods of this value. Once this value is computed, Ag_a decides to consider Ag_b trustworthy or not depending on the threshold W_{Ag}.

IV. Trust Computing

A. Method 1

To compute trust in our model, we propose a probabilistic method by investigating the distribution of the random variable X representing the trustworthiness of Ag_b. Let us first consider the simple case where X takes only two values: 0 (the agent is not trustworthy) or 1 (the agent is trustworthy). Therefore, variable X follows a *Bernoulli distribution* $\beta(1,p)$. Accordingly $E(X) = p$ where $E(X)$ is the expectation of the variable X and p is the probability that the agent is trustworthy. Here, p is the probability we are looking for. Therefore it is enough to evaluate the expectation $E(X)$ to find $Tr_{Ag_a}^{Ag_b}$. However, this expectation is a theoretical mean that we must estimate. To this end, we can use the *Central Limit Theorem (CLT) and the law of large numbers*. The CLT states that whenever a sample of size n $(X_1,\ldots,X_n)$ is taken from any distribution with mean μ, then the sample mean $(X_1+\cdots+X_n)/n$ will be approximately normally distributed with mean μ. As an application of this theorem, the arithmetic mean (average) $(X_1+\cdots+X_n)/n$ approaches a normal distribution of mean μ, the expectation and standard deviation $\sigma/\sqrt{n}$. Generally, and according to *the law of large numbers*, the expectation can be estimated by the weighted arithmetic mean.

Our random variable X is the weighted average of n independent variables X_i that correspond to Ag_b's trust level according to the point of view of *trustworthy agents* Ag_i and *referee agents* Rf_j. These variables follow the same *Bernoulli distribution*. They are also independent because indeed the probability that Ag_b is trustworthy according to an agent Ag_{t1} is independent of the probability that this agent (Ag_b) is trustworthy according to another agents Ag_{t2}. Consequently, the variable X follows a normal distribution whose average is the weighted average of the expectations of the independent variables X_i. The mathematical estimation of expectation $E(X)$ is given by the following equation:

$$M_0=\frac{\sum_{i=1}^{n}\left(Tr_{Ag_a}^{Ag_i}\times Tr_{Ag_i}^{Ag_b}\right)+\sum_{j=1}^{m}\left(Tr_{Ag_a}^{Rf_j}\times Tr_{Rf_j}^{Ag_b}\right)}{\sum_{i=1}^{n}Tr_{Ag_a}^{Ag_i}+\sum_{j=1}^{m}Tr_{Ag_a}^{Rf_j}} \quad (2)$$

The number of requested references (m) is defined by Ag_a and is related to the number n of *trustworthy agents* Ag_a has in order to ensure that enough number of third parties have been involved to participate in the evaluation. If Ag_b cannot (or refuse to) provide the requested number of referees, this will negatively affect his trust evaluation process, particularly if the number of trustworthy agents involved in this process is not enough.

In order to introduce its *referee agents*, Ag_b forwards Ag_a's information to each one of referees he wants to introduce. *Referee agents* forward directly to Ag_a the trust level of Ag_b according to their past experience of direct interaction with Ag_b. Categorizing the *referee agents*, there are three possibilities: (1) the *referee agent* is also a *trustworthy agent* of Ag_a, which gets more priority as long as his suggestion about Ag_b would be considered to be more important and thus Ag_a will give more trustworthiness value to this particular agent; (2) Ag_a knows the *referee agent* (there is assigned trust level for the *referee agent* in Ag_a's part) and he considers the *referee agent*'s suggestion by including his trustworthiness value from Ag_a's point of view based on previous reputation that the referee has made; and (3) Ag_a does not know the *referee agent*, in which case Ag_a can adopt different policies. He can just accept the *referee agents* to whom he had direct experience and thus the corresponding assigned trustworthiness value is used. In this policy, *referee agents* with no interaction history with Ag_a will be automatically removed. However Ag_a may take the policy of assigning a default value η gained by overall reputation of such *referee agents* and start over. The value η is specific for each *referee agent*. Therefore we can advance equation 1 in order to make the consulting agents' trust estimation more flexible:

$$Tr_{Ag_a}^{Rf_j}=\begin{cases}\dfrac{\sum_{i=1}^{m}v_i w_i NI_{i\,Ag_a}^{Rf_j}}{\sum_{i=1}^{n}v_i w_i NI_{i\,Ag_a}^{Rf_j}}, & \text{n>0;}\\ \eta, & \text{n=0.}\end{cases} \quad (3)$$

In this equation if the assigned value η is still 0, that means Ag_a does not know the introduced *referee agent* at all (even Ag_a could not find a basis reputation for the agent that it could estimate the *referee agent*'s credibility). In this case Ag_a does not consider his suggestion about Ag_b, but he saves the referee's suggestion anyway in order to compare it with the real behavior Ag_b performs after starting interaction with Ag_a. Thus the referee is known by Ag_a from now on and his trust level is calculated by the adjustment of the final answer for Ag_b and the referee's first suggestion.

B. Method 2

The value M_0 represents an estimation of $Tr_{Ag_a}^{Ag_b}$. This estimation, however, does not take into account the *number of interactions* between the *trustworthy/referee agents* and Ag_b. These numbers are important factors because they promote information coming from agents knowing more about Ag_b. The agents who had high number of interactions with Ag_b are considered as good sources of information about his trustworthiness (although there may be some agents who had not very high number of interactions but they are accurate enough that provide very precise information about other agents). In addition, another factor might be used to reflect the *timely relevance* of transmitted information. This is because the agent's environment is dynamic and may change quickly. The idea is to promote recent information and to deal with out-of-date information with less emphasis. The timely relevance could be represented as a coefficient when computing the agent's trust. In our model, we assess the factor $TR(\Delta t_{Ag_i}^{Ag_b})$ by using the function defined in equation 4. We call this function:

the *Timely Relevance* function.

$$TR(\Delta t_{Ag_i}^{Ag_b}) = e^{-\lambda\ ln(\Delta t_{Ag_i}^{Ag_b})} \quad (4)$$

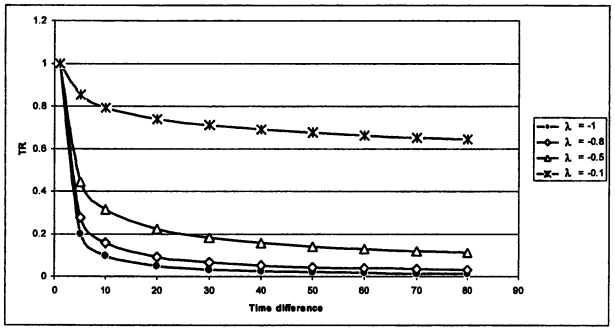

Fig. 3. Timely relevance behavior.

Δt is the time difference between the current time and the time at which Ag_i updates its information about Ag_b's trust. λ is an application-independent coefficient. The intuition behind this formula is to use a function decreasing with the time difference like what is shown in figure 3. Consequently the more recent the information is, the higher is the timely relevance coefficient. The function *ln* is used for the computational reasons when dealing with large numbers. In fact, this function is similar to the well known *reliability function* for systems engineering ($R(t) = e^{-\lambda t}$). The value λ is to be set appropriately. The time is more important when λ is close to -1. If the number of available *trustworthy and referee agents* is big enough, it will be wise to consider the time relevance. However, if this number is small, so all the information should be considered even the old one. Consequently, the time relevance in this case could be disregarded. To make this idea concrete, λ could be evaluated using equation 5 as follows:

$$\lambda = -\frac{n+m}{n_{max}+m_{max}} \quad (5)$$

The negative sign in this equation is needed because of the decreasing exponential curve. The n and m denote the number of available *trustworthy* and *referee agents* Ag_a has to get information from. n_{max} and m_{max} denote the maximum number of the *trustworthy* and *referee agents* a typical agent needs. Therefore Ag_a makes a balance in order to set up a more accurate and adequate provided decisions.

Equation 6 gives us an estimation of $Tr_{Ag_a}^{Ag_b}$ if we take into account these factors. This equation is composed of two different terms representing the values got from two different consulting communities involved in trust evaluation. The function Ω is defined as the summation of the trust values estimated by the *trustworthy agents* together with their related self-trustworthiness, timely relevance and the number of interactions $N_{Ag_b}^{Ag_i}$ between the *trustworthy agents* Ag_i and the target agent Ag_b. The Ψ function indicates the similar relative coefficients regarding to the corresponding *referee agents*.

$$Tr_{Ag_a}^{Ag_b}(\Delta t) = \frac{\Omega(Ag,n,Ag_a,Ag_b)+\Psi(Rf,m,Ag_a,Ag_b)}{\Omega'(Ag,n,Ag_a,Ag_b)+\Psi'(Rf,m,Ag_a,Ag_b)} \quad (6)$$

where

$$\Omega(Ag,n,Ag_a,Ag_b) = \sum_{i=1}^{n} Tr_{Ag_a}^{Ag_i} \times Tr_{Ag_i}^{Ag_b} \times TR(\Delta t_{Ag_i}^{Ag_b}) \times N_{Ag_b}^{Ag_i} \quad (7)$$

$$\Omega'(Ag,n,Ag_a,Ag_b) = \sum_{i=1}^{n} Tr_{Ag_a}^{Ag_i} \times N_{Ag_b}^{Ag_i} \times TR(\Delta t_{Ag_i}^{Ag_b}) \quad (8)$$

$$\Psi(Rf,m,Ag_a,Ag_b) = \sum_{j=1}^{m} Tr_{Ag_a}^{Rf_j} \times Tr_{Rf_j}^{Ag_b} \times TR(\Delta t_{Rf_j}^{Ag_b}) \times N_{Ag_b}^{Rf_j} \quad (9)$$

$$\Psi'(Rf,m,Ag_a,Ag_b) = \sum_{j=1}^{m} Tr_{Ag_a}^{Rf_j} \times N_{Ag_b}^{Rf_j} \times TR(\Delta t_{Rf_j}^{Ag_b}) \quad (10)$$

There is a possibility that the *referee agent* is untrustworthy from Ag_a's point of view. That means Ag_a would rate a negative value to Rf_j as a result of bad past direct experience Ag_a had with Rf_j. Consequently Rf_j's suggestion could not be helpful in the evaluation process of Ag_b and thus Ag_a may decide not to consider it at all (set flag F to 0). However it would be reasonable if Ag_a also considers Rf_j's suggestion about Ag_b and consequently decreases the Ag_b's credibility as he introduced a bad recommender (set flag F to 1). Therefore we advance the Ψ function by replacing the $Tr_{Ag_a}^{Rf_j}$ by a new overall trust evaluation value $Tr_{Ag_a}^{''Rf_j}$. Equations 11 and 12 identify the value of $Tr_{Ag_a}^{''Rf_j}$ as follows:

$$Tr_{Ag_a}^{''Rf_j} = \begin{cases} Tr_{Ag_a}^{Rf_j}, & Tr_{Ag_a}^{Rf_j} \geq 0; \\ Tr_{Ag_a}^{'Rf_j}, & Tr_{Ag_a}^{Rf_j} < 0. \end{cases} \quad (11)$$

$$Tr_{Ag_a}^{'Rf_j} = \begin{cases} 0, & \text{F=0;} \\ Tr_{Ag_a}^{Rf_j} \times F, & \text{F=1 and } |Tr_{Ag_a}^{Rf_j}| > \lambda; \\ -\lambda \times F, & \text{F=1 and } |Tr_{Ag_a}^{Rf_j}| < \lambda. \end{cases} \quad (12)$$

Here the $Tr_{Ag_a}^{''Rf_j}$ would be $Tr_{Ag_a}^{Rf_j}$ itself if the *referee agent* has a positive credibility from Ag_a's point of view, but we would give $Tr_{Ag_a}^{'Rf_j}$ value to the overall referee's trust level if it has a negative credibility in Ag_a's side. Consequently $Tr_{Ag_a}^{'Rf_j}$ would be evaluated by three cases. If the flag F is set to 0, that means the negative creditable *referee agents* would not be considered, but if the flag is 1, Ag_a will consider the *referee agent*'s suggestion and respectively overall it would make a negative value. That will affect the trust evaluation of the Ag_b while it would be decreased. Ag_b will get this penalty because he introduced a bad *referee agent*. But we define a limit here that guarantees a least penalty an agent would get in such cases. That means if the absolute trustworthiness value assigned to *referee agent* was less than a threshold λ then overall it would not affect that much and as a results agents may not care to introduce bad agents. Then in such cases the value λ would be considered to provide the appropriate negative trust measurement.

We denote here that removing $TR(\Delta t_{Ag_i}^{Ag_b})$ from the Equation 6, results in the classical probability equation used to calculate the expectation $E(X)$. Equation 6 takes into account the four most important factor: (1) the trustworthiness of *trustworthy/referee agents* according to the point of view of Ag_a ($Tr_{Ag_a}^{Ag_i}$ and $Tr_{Ag_a}^{Rf_j}$); (2) the Ag_b's trustworthiness

according to the point of view of *trustworthy/referee agents* ($Tr_{Ag_i}^{Ag_b}$ and $Tr_{Rf_j}^{Ag_b}$); (3) the number of interactions between these *trustworthy/referee agents* and Ag_b ($TN_{Ag_b}^{Ag_i}$ and $TN_{Ag_b}^{Rf_j}$) and (4) the timely relevance of information transmitted by *trustworthy/referee agents* ($TR(\Delta t_{Ag_i}^{Ag_b})$ and $TR(\Delta t_{Rf_j}^{Ag_b})$), as communicated by Ag_i to Ag_a following the strategies previously indicated.

This Equation shows how trust can be obtained by merging the trust values transmitted by *trustworthy agents*. This merging method takes into account the proportional relevance of each trust value, rather than treating them equally. To compute this trust, the relevant information a *trustworthy agent* Ag_i should provide to Ag_a (i.e. the content of Tr_Inf) are: 1) the total number of interactions Ag_i had with Ag_b ($TN_{Ag_b}^{Ag_i}$); 2) the number and time of recent interactions between them ($N(\Delta t_{Ag_i}^{Ag_b})$ and $\Delta t_{Ag_i}^{Ag_b}$); 3) and the overall Ag_i's evaluation of Ag_b's trust ($Tr_{Ag_i}^{Ag_b}$), likewise for the *referee agent* Rf_j.

V. Model Properties

The proposed framework generally lies on rational trust estimation of one agent about others. In this section we tend to move to probabilistic point of view of such framework. In order to clarify the analysis of situations might happen in more details, we start over by some definitions:

Definition 2: Let $\mathcal{T}_{Ag_a}$ and $\mathcal{R}_{Ag_a}$ respectively be the set of *trustworthy agents* of Ag_a and *referee agents* that are proposed to Ag_a. Let $N_T = \sum_{Ag_i \in \mathcal{T}_{Ag_a}} N_{Ag_b}^{Ag_i}$ and $N_R = \sum_{Rf_j \in \mathcal{R}_{Ag_a}} N_{Ag_b}^{Rf_j}$. N_T and N_R denote the cumulative number of interactions done in the *trustworthy* and *referee communities* with Ag_b.

Definition 3: Let T_b be the overall estimation of $Tr_{Ag_a}^{Ag_b}$ calculated from equation 6. We set $T_i, (i = 1..|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|)$ to be the overall Ag_i's contribution in the $Tr_{Ag_a}^{Ag_b}$ trust evaluation process. Intuitively $T_b = \sum_{i=1}^{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|} T_i$. Therefore the corresponding contribution percentage of each agent would be $\alpha_i = \frac{T_i}{T_b}$.

Here Ag_i's contribution is taken as a positive value. We formulate the probability density function (PDF) as commutative probability of α_is as they lay all in the range [0,1] and the $\sum_{i=1}^{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|} \alpha_i = 1$. Intuitively α_is form a uniform distribution over $[0,1]$. Therefore we take α_i as the probability of Ag_i's contribution which lies in $[0,1]$. For example if the high contribution is assumed to be at least 0.6, we get a distribution that is 0 over $[0, 0.6]$ and 2.5 over $[0.6, 1]$ or if we are looking for a particular range of contribution which is between 0.2 and 0.4, the distribution is 0 over $[0, 0.2]$ and $[0.4, 1]$ and 5 over $[0.2, 0.4]$.

Assume function $f : [0,1] \rightarrow [0,1]$ defines the distribution of α. Therefore we get $\int_0^1 f(\alpha_i)d\alpha = 1$ satisfying the probability density function. Consequently looking for a probability which lies in the range $[\beta_1, \beta_2]$ strictly defined for a particular Ag_i would be $\int_{\beta_1}^{\beta_2} f(\alpha_i)d\alpha_i$. The mean value of f is $\frac{\int_0^1 f(\alpha_i)d\alpha}{1-0} = 1$ and the standard deviation is $\delta = \frac{\int_0^1 [f(\alpha_i)-T_b]^2 d\alpha}{1-\alpha_i}$.

Property 1: The standard deviation decreases as the quality of relation get more robust. That means the higher $N_T + N_R$, the more interactions done and therefore more contribution from different agents are expected. This also refers to Bayesian distribution of assuming an equiprobable prior. So the uniform distribution gets more as more general information is provided.

Property 2: When agents keep interacting and get more accurate knowledge about each other, N_T and N_R are supposed to have high values after a certain amount of elapsed time that leads release of more accurate trust estimation from consulting agents.

Remark 1: It is reasonable to assume $N_R > N_T$ as Ag_b better knows $\mathcal{R}_{Ag_a}$, so they had more interactions with Ag_b than $\mathcal{T}_{Ag_a}$.

Definition 4: Let $\widehat{\alpha}$ be the mean value of α_is set as: $\widehat{\alpha} = \frac{\sum_{i=1}^{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|} \alpha_i}{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|}$ and let $\mathcal{T}'_{Ag_a} = \{Ag_i | Ag_i \in \mathcal{T}_{Ag_a}, \alpha_i > \widehat{\alpha}\}$ and $\mathcal{R}'_{Ag_a} = \{Ag_i | Ag_i \in \mathcal{R}_{Ag_a}, \alpha_i > \widehat{\alpha}\}$ represent the set of *trustworthy* and *referee agents* which perform higher contribution.

Property 3: the more number of interactions, the more accurate evaluation process takes place. Therefore we get higher elements in $\mathcal{T}_{Ag_a}$ as they contribute more in evaluation process and we tend to observe more contribution from *trustworthy agents* as they are already known by Ag_a and there is a less risk of fake estimation. In general the use of referrals are mostly in the absence of the *trustworthy agents*. Therefore the proposed model tends to get more contribution from the *trustworthy agents* and the *referee agents* with the high credibility and reputation rather than a low credibility agent or a malicious one. In section VII we try to minimize the estimation error by pushing the contribution percentage values to get more contribution of the trustworthy and high credibility agents.

VI. Refusal Penalty

As specified in section IV there is a dialogue game $Rep\_NotHave(Ag_i, Ag_a, *, t')$ used in case the contributing agent declare no estimation regarding to Ag_b's trust level. There is a possibility that a malicious agent try to deviate the evaluation process. In order to avoid this issue and penalize the agents who lie about not having the requested information, at the time $t_4 > t$" (after getting all information from others) Ag_a requests Ag_b about the number of interactions he had with each one of the requested agents included in the message. Therefore Ag_a uses the following dialogue game to request:

$$\forall Ag_i \in \mathcal{T}_{Ag_a}, Req\_Inf(Ag_a, Ag_b, N_{Ag_b}^{Ag_i}, t_4)$$
$$\forall Rf_i \in \mathcal{R}_{Ag_a}, Req\_Inf(Ag_a, Ag_b, N_{Ag_b}^{Rf_j}, t_5)$$

and Ag_b in return replies to Ag_a by providing the corresponding number of interactions using the following dialogue games. Figure 4 is the modified version of the figure 2 highlighted in gray color.

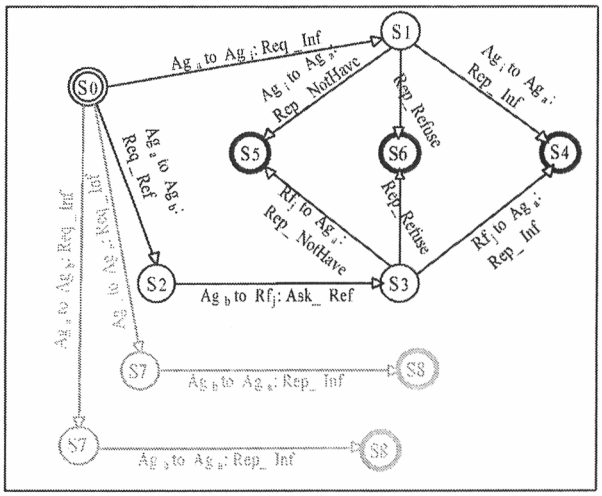

Fig. 4. Protocol of gathering information from *trustworthy* and *referee agents*

$$Req\_Inf(Ag_a, Ag_b, N_{Ag_b}^{Ag_i}, t_4) \Rightarrow Rep\_Inf(Ag_b, Ag_a, N_{Ag_b}^{Ag_i}, t_6)$$

$$Req\_Inf(Ag_a, Ag_b, N_{Ag_b}^{Rf_j}, t_5) \Rightarrow Rep\_Inf(Ag_b, Ag_a, N_{Ag_b}^{Rf_j}, t_7)$$

Now Ag_a is able to define which agents who refused to release information were trying to confuse others. In this case depending of the type of the agent, Ag_a makes an adjustment in the trustworthiness of the refusing agent. Let R_t and R_r denote the predefined penalty percentage Ag_a deduces from the credibility of the agent who has been noticed to have different number of interactions reported by Ag_b. Obviously $R_t > R_r$ as *trustworthy agents* are not supposed to lie from Ag_a's point of view. Therefore assume Ag_r is the agent who is to get reduction. Equation 13 shows the reduction credibility from Ag_a's side.

$$Tr_{Ag_a}^{Ag_r} = \begin{cases} Tr_{Ag_a}^{Ag_r} \times R_t, & \text{if } Ag_r \in \mathcal{T}_{Ag_a}; \\ Tr_{Ag_a}^{Ag_r} \times R_r, & \text{if } Ag_r \in \mathcal{R}_{Ag_a}. \end{cases} \quad (13)$$

VII. Maintenance

Generally in trust evaluation we try to minimize the adverse affects the consulting agents may produce. For instance, two agents who have a strong relationship can support each other in trust evaluation and overestimate their trust level when they have been introduced as *referee agents*. Although the relationship strengthen ratio can be certainly inserted as a trust measure value to increase the accuracy of the *referee agent*, it is not good as a generic basis and thus we characterize our solution on the number of interactions done between two agents. That implicitly means Ag_a can expect a more accurate suggestion from a *referee agent* who had a large number of interactions with Ag_b comparing to the *referee agent* who had less. Thus we should try to give more emphasis to such agents that previously had large number of interactions by Ag_b in terms of accepting their idea. Respectively these agents should affect more when the opposite of their suggestion turned out to be true.

Therefore Ag_a needs to perform a maintenance to adjust the consulting agents credibility. Generally Ag_a is more confident about its *trustworthy agents* as they have shown an acceptable trustworthiness so far, but the *referee agents* are chosen by Ag_b, so we should always consider the possibilities like the cooperating partners may vote in favor of each other or competing agents may underrate their opponents. Therefore Ag_a after a period of interacting with Ag_b performs maintenance in order to evaluate the witness reputation to assess the consulting agents' trust level. In trust evaluation process done by Ag_a maybe there were some *referee agents* involved in but their suggestion were not took into account as they were not known by Ag_a and consequently not eligible to interfere. But Ag_a did not discard their suggestion; after the maintenance Ag_a would be able to estimate such *referee agents*' credibility as long as they are known (because of their referee history) by Ag_a from now on. The rational behind this maintenance is to compare the actual behavior Ag_b performed after starting interaction with Ag_a with the suggestions provided about Ag_b's credibility by others.

Now if Ag_a received a reference from *referee agent* Rf_j (this accuracy check is not just specified to *referee agents*; *trustworthy agents* are also checked), Ag_a then adjusts Rf_j's trust level by comparing the actual performance of Ag_b, as a result of a period of direct interaction experience, and what Rf_j provided as the suggested trust level for Ag_b. Therefore thresholds ν_T and ν_R are associated as inaccuracy tolerance thresholds for the *trustworthy* and *referee agents*. We assign two different thresholds because the *referee agents* were supposed to deliver a more accurate information about Ag_b comparing to the *trustworthy agents* because they have been introduced by Ag_b. By doing so, if the difference is greater than the associated threshold for the agent, the consulting agent's trust level should be dropped to some extent, otherwise it will be enhanced regarding to the importance of suggestion provided, this value ϕ is defined by Ag_a. The important thing here is the ratio of dropping down the trustworthiness value of the consulting agent after the comparison. Let us assume the trust level assigned by Ag_a to Ag_b is $Tr_{Ag_a}^{Ag_b}$ and the value provided by the *referee agent* is $Tr_{Rf_j}^{Ag_b}$. Therefore Ag_a adjusts the trustworthiness level of the agent Rf_j by the following equation:

$$Tr_{Ag_a}^{Rf_j} = \begin{cases} Tr_{Ag_a}^{Rf_j} - N_{Ag_b}^{Rf_j} \times D_R, & \text{if } D_R > \nu_R; \\ Tr_{Ag_a}^{Rf_j} + \phi, & \text{if } D_R < \nu_R. \end{cases}$$

$$D_R = |Tr_{Rf_j}^{Ag_b} - Tr_{Ag_a}^{Ag_b}|; \quad (14)$$

The value D_R defines the inaccuracy of the trust level regarding to the specific consulting agent. The inaccuracy is checked by the predefined threshold (here for *referee agents* ν_R) to recognize whether the *referee agent*'s suggestion was

apart from the real value. If Ag_a, after comparison, considers the *referee agent* trustworthy, it increases the current trust level by the value ϕ, otherwise it decreases the trust level by the ratio related to the corresponding number of interactions done by the *referee agent* Rf_j and Ag_b. The number of interaction is used as a measure here as we assume the higher number of interactions, the more accurate information supposed to perform, consequently the more decrease when wrong information is provided. Therefore having recorded the ratings provided by agent R about other agents, Ag_a can evaluate or adjust R's credibility after performing maintenance and checking the differences. Obviously the ratio of adjustment is not very high and affective to the *trustworthy agents* as they were not supposed to know Ag_b and thus provide an accurate information about him. However there may be a good increase in the trust level provided by Ag_a to the consulting agents regarding because of their accuracy in providing information about Ag_b.

Strictly speaking, the objective is to get the most accurate evaluation possible, so let us rephrase the objective as following optimization problem:

Definition 5 Let $T_b = \sum_{i=1}^{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|} \mathcal{M}_i \mathcal{T}_i$ in which $\mathcal{T}_i$ is the suggested trustworthiness level of Ag_b provided by Ag_i (either trustworthy or referee) and $\mathcal{M}_i$ denotes the combination of all measures considered to give weight to the suggestion of the agent in question.

To minimize the error of the obtained wights from the equation 6, we should consider the best sequence of weights that could minimize the difference of the estimated trust value and the actual behavior of Ag_b regarding to the sequence of the weights assigned for each consulting agent, shown as $\underline{\mathcal{M}}$. The corresponding optimization problem is shown in equation 15:

$$min_{\underline{\mathcal{M}}} \sum_{i=1}^{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|} |\mathcal{M}_i \mathcal{T}_i - T_b| \tag{15}$$

Due to the fact that Ag_a only gives one chance to the contributing agents to release their suggestion, T_is and T_b are assumed to be constant. Therefore by the information provided, the vector $\underline{\mathcal{M}}$ with the size $(|\mathcal{T}_{Ag_a}| + |\mathcal{R}_{Ag_a}|) \times 1$ denotes the computed measurement of consulting agents.

As T_is are assumed coefficients, the expression $\underline{\mathcal{M}}.\underline{\mathcal{T}}^T - T_b$, in which $\underline{\mathcal{T}}^T$ is the transposed vector of the suggested trust values, would be a linear expression as for each variable $\mathcal{M}_i$, there is a corresponding coefficient multiplied, therefore we can perform one iteration of the *steepest descent* minimization method to get the appropriate direction toward the minimum, together with the proper step size for moving the initially obtained $\underline{\mathcal{M}}$ to $\underline{\mathcal{M}}'$ which would be close enough to the minimum possible. Therefore $\underline{\mathcal{M}}'$ can be assumed as the best measurement Ag_a could have performed in that situation. The idea is for each consulting agent Ag_i, checking the $|T_i - T_b|$ in case of big difference decreases the corresponding $\mathcal{M}_i$ (consequently the trustworthiness of the related agent) and in the opposite in case of very minor difference, it increases the corresponding $\mathcal{M}_i$. Then the new values can be set as the constraints by which the minimization should be performed subject to the defined constraints.

Assume $Ag_i \in \mathcal{T}_{Ag_a}$ is a *trustworthy agent*. Therefore regarding to the predefined inaccuracy threshold ν_T we would face two cases of being less or more than ν_T. Thus in case of less, it means the suggested value is quite close to the actual performance, therefore the Ag_i should get more emphasis regarding to its credibility from Ag_a's point of view, equation 16. Therefore if the already measured $\mathcal{M}_i$ is less than the corresponding element in the minimized sequence, $\mathcal{M}'_i$, the credibility of Ag_i which is included in $\mathcal{M}_i$ should be increased by the predefined ratio $\psi_T > 1$. This ratio shifts the weight $\mathcal{M}_i$ toward $\mathcal{M}'_i$. We did not replace $\mathcal{M}_i$ by the $\mathcal{M}'_i$ because in the minimization there are other constraints contributing that all together tend to get closer to the minimum point which satisfies with respect to other upcoming constraints. If the already measured $\mathcal{M}_i$ is greater than the corresponding element in the minimized sequence, $\mathcal{M}'_i$, this means the overall balance of the measurements tends to have less value in $\mathcal{M}_i$, but as long as Ag_i had a quite close suggestion provided, we set up a new constraint for the next minimization run that the value $\mathcal{M}_i$ should be greater than the current value of $\mathcal{M}_i = m_i$ to avoid this mistake. Therefore for both cases (less or more than the inaccuracy threshold) we save the current value of $\mathcal{M}_i$ as m_i and insert the new constraint $\mathcal{M}_i > m_i$ into the list of the constraint in the minimization problem of equation 15.

$$|T_i - T_b| < \nu_T \Rightarrow \begin{cases} Tr_{Ag_a}^{Ag_i} = Tr_{Ag_a}^{Ag_i} \times \psi_T, & \text{if } \mathcal{M}_i < \mathcal{M}'_i; \\ m_i = \mathcal{M}_i, & \text{if } \mathcal{M}_i > \mathcal{M}'_i. \end{cases}$$

$$constraint\ \mathcal{M}_i > m_i; \tag{16}$$

In the case of error more than the inaccuracy threshold, equation 17, again we may observe that the already measured $\mathcal{M}_i$ is less than the corresponding element in the minimized sequence, $\mathcal{M}'_i$. This means the minimized combination of weights tends to have more value for the $\mathcal{M}_i$ which we do not want because the estimation difference is high, therefore we set up the corresponding constraint that the $\mathcal{M}_i$ should be less than the current value m_i. Respectively if the $\mathcal{M}_i$ is greater than $\mathcal{M}'_i$, means we should decrease $\mathcal{M}_i$ in terms of the credibility of Ag_i to get closer to $\mathcal{M}'_i$.

$$|T_i - T_b| > \nu_T \Rightarrow \begin{cases} m_i = \mathcal{M}_i, & \text{if } \mathcal{M}_i < \mathcal{M}'_i; \\ Tr_{Ag_a}^{Ag_i} = Tr_{Ag_a}^{Ag_i} / \psi_T, & \text{if } \mathcal{M}_i > \mathcal{M}'_i. \end{cases}$$

$$constraint\ \mathcal{M}_i < m_i; \tag{17}$$

$$min_{\underline{\mathcal{M}}} \sum_{i=1}^{|\mathcal{T}_{Ag_a}|+|\mathcal{R}_{Ag_a}|} |\mathcal{M}_i \mathcal{T}_i - T_b|$$

subject to

List of constraints;

As the new constraints are inserted, a new run will be performed which is more restricted comparing to before; after few runs finally we settle down with a balance for the rest of the agents by which Ag_a can adjust their $\underline{\mathcal{M}}$ as $\underline{\mathcal{M}}'$.

VIII. Proof of Concepts

In this section we assess the model efficiency and implement a proof of concept prototype. In this prototype, 70 agents are implemented as $Jadex^{©TM}$ agents, i.e. they inherit from the basic class $Jadex - Simulator^{©TM}$ *Agent*. The agent reasoning capabilities are implemented as Java modules using logic programming techniques. As Java classes, agents have private data called *Belief Data*. The different dialogue games are given by a data structure and implemented using tables and the different actions expected by an agent in the context of a particular dialogue game are given by a table called *data_representative_manager*. The different agents' reputation values that an agent has are recorded in a data structure called *data_reputation*. Each agent has also a knowledge base about the reputation of other agents, called *table_reputation*. Such a knowledge base has the following structure: $Agent - name$, $Agent - reputation$, $Total-interaction-number$, $Recent-interaction-time$ and $Recent - interaction - number$. The visited agents during the evaluation process and the agents added in the reputation graph are recorded in two $Jadex^{©TM}$ $beliefsets$ called: *table_visited_agents* and *table_graph_reputation*. Fig.5 Illustrates these different data structures.

Fig. 5. The data structures of the prototype

The main steps of the evaluation process of Ag_b's reputation are implemented as follows:

1) Ag_a consults his knowledge base *data_reputation* of type *table_reputation* and sends a request to his *trustworthy agents* Ag_i (i = 1,.., n) about Ag_b's reputation. A same request is sent to the referee agents introduced by Ag_b. The $Jadex^{©TM}$ primitive *Send* makes it possible to send the request as a messages that we call *Ask_Reputation* of *MessageEvent* type. Ag_a sends this request starting by confidence agents whose reputation value is the highest.
2) In order to answer the Ag_a's request, each agent Ag_i executes a plan instance that we call *Plan_ev_Ask_Reputation*. Thus, using his knowledge base, each agent Ag_i offers to Ag_a an Ag_b's reputation value if Ag_b is known by Ag_i. If not, Ag_i proposes a set of *trustworthy agents* from his point of view, with the relevant information discussed above. Referee agents do not execute this additional plan.
3) When Ag_a receives the *Reputation_Value* message, he executes a plan: *Plan_ev_Reputation_Value*. According to this plan, Ag_a adds to a graph structure called *graph_data_reputation* two information: **1)** the agent Ag_i and his reputation value as graph node; **2)** The reputation value that Ag_i offers for Ag_b, the number of times that Ag_i interacted with Ag_b, the time period of the recent interactions, and the number of these interactions as arc relating the node Ag_i and the node Ag_b.
4) Steps 1, 2, and 3 are applied again by substituting *data_reputation* by *new_data_reputation*, until all the consulted agents offer a reputation value for Ag_b or until one of the two fixed limits is reached.
5) Evaluate the Ag_b's reputation value using the information recorded in the structure *graph_data_reputation* by applying Equation 6.

IX. Related Work

Perhaps the best-known approaches to trust in multi-agent systems are FIRE [9], ReGret [20] and Referral [24]. In this section in addition we get more into details by analyzing some recently emerged systems like SPORAS [28], Formal [16], HIT [21], Adaptive [29] and Statistics [30]. So far the proposed approaches are distinguishable by the following classifications: 1) Policy-based trust; 2)Reputation-based trust; 3) General model of trust; 4)Trust in information resources. Generally speaking all the approaches are following a direction to overcome the following problems: The model should be provided by adequate information related to the environment and the contributing agents; they tend to avoid consulting with a central control unit who is always subject to single point of failure or huge bottleneck (for example in online auction development). Agents are aimed to make estimation independently; there are always malicious agents who try to distract the overall process; they can either try to slander other agents by lying about its trust level or supporting an agent on purpose, try to exaggerate about its credibility.

Reputation-based trust is mostly under analysis. It is worth mention the most recent research areas in reputation-based trust model are as follows: a)Interaction trust which would be based on the direct interaction of two parties and provided services; b) Trust based on the type of prior interactions; c)Witness reputation which would be based on the reports provided by the third parties; and d) Certified reputation which would be based on the references requested from some agents to report their belief about a particular agent's behavior.

Recently some online trust models have been developed (see [7] for a detailed survey). The most widely used are those on *eBay* and Amazon Auctions and also in *Virtual Worlds* [22] Applications and *Stock Markets* [23]. Both of these are implemented as a centralized trust system. One of the

substantial characteristics of *eBay* is that the transactions done every moment are not saved by formal contractual guarantees. But buyers rely on the trust which is based on the simple ratings previously provided to sellers as feedback. Likewise, the buyer/seller overall rate the other party's cooperation as feedback. Therefore the history of trader's past ratings is exposed to entire community. Thus reputation in these models are not very reliable. In addition, these models are not suitable for applications in open Multi-Agent Systems such as agent negotiation because they are too simple in terms of their trust rating values and the way they are aggregated.

FIRE model proposed by Huynh, Jennings and Shadbot [9], solves the problem of collecting the required information by the evaluator to assess the trust of his partner. In a model-based on witnesses there is a possibility for witness to refuse sharing their experiences. Therefore they propose a method called certified reputation. In this method they add an additional factor for defining the trustworthy of *referee agents* which are introduced by the target agent. The most important aspect of this method is that an agent quickly evaluates the targets trust value, because of the small number of interactions needed while it does not create the trust graph. In some cases agents do not propose a good *referee agent* and as a rational agent, it picks up the referee who is more beneficial for him rather than the system, thus in this case the final trust rate would be affected with non-reliable information about the target agent. Eventually the agents imagination about the target agent will not be true, therefore the evaluating agent has to evaluate the *referee agents*, although it will cost an extra computational overhead for the method. Eventually the trust graph that we demonstrate in this paper can overcome this limitation and reduce the overhead.

The idea of witness reputation has been used by Sabater who proposed a decentralized trust model [20] called ReGret. He used the reports from the witnesses in addition to the technique based on direct interaction experience. One of the substantial aspects of this work is unlike the previous approaches, the rating are dealt according to their recently relevance. Thus, old ratings are given less importance compared to new ones. Sabater's work is sensitive to noise and thus vulnerable as it does not represent witness locations. Also, it does not notice distractions made by some malicious agents. In our model, the issue is managed by considering the witnesses trust and our merging method takes into account the proportional relevance of each reputation value, rather than treating them equally.

Yu and Singh [24], [25], [26] by applying social network concepts in multi-agent systems proposed a new trust model called Referral. In this method witness agents use message passing method for transmitting information. Doing so they retrieve ratings among social networks. An aspect of their method is similar to the role of links that search engines use to obtain a web page, approaching to another source of information. In our model we use a similar graph model to TrustNet, although our proposed trust model has many differences as we use argumentation-based negotiation so that agents use argumentation-based reasoning whereas the referral does not support any particular reasoning. In addition the possibility of having an agent who may lie has not taken into account. Overall we cannot use referral model in a dynamic environment because the time relevance is not considered in the trust graph.

SPORAS is also another system which perform simple rating. These systems [28] suffer from rating noise because they treat all ratings equally. Consequently some new approaches emerged to include some measurements related to the trust level of particular agent. In addition SPORAS is a centralized approach so it is not suitable for open systems.

Singh in the other work with Wang developed as algebra [16] for aggregating trust over graphs understood as webs of trust. They believe current approaches for combining trust reports tend to involve ad hoc formulas. So they bring up a solution that regulates the difficulty of understanding from a conceptual bases which is the concept of discounting. In their work dynamism is accommodated by discounting over time and composition by discounting over the space source. They have developed a principled evidential trust model that would underlie any such agent system where trust reports are gathered from multiple sources.

Regarding to ad hoc formulation, Singh's similar work has been done by Velleso who assigns trust levels in ad hoc network [21]. The aspect of their work is that they have refereed to human concept of trust. Similar to our work they use the recommendations by *trustworthy agents* in addition to their own direct experience. They tried to balance the recommendations regarding to recently relevance and relationship maturity, but the agents do not have reasoning capabilities, moreover they do not have policies taken for dealing with the malicious agents.

Song, Phoha and Xu, proposed an Adaptive recommendation trust model [29] for multi-agent systems. They design a neural network for evaluating multiple recommendations of various trust standards with and without deceptions. They used an ordered depth first search (DFS) for delaying the first initial set of qualified recommendations (preparing a proper data set for proposed neural network input). In the second stage they design a neural network which is based on back propagation. The output of this stage will be the actual set of qualified recommendations. The most important advantage of this model is adaptively and flexibility that captures the dynamic nature of online trust. On the other hand using neural network in dynamic environment needs much more time for training faze of neural net, thus when our input data set has changed our designed neural net must be adapted and it needs a large amount of time considering time period for each iteration in Multi-Agent Systems. As each trust model needs to update its recommendations and we have to consider the time relevance factor in recommender qualification faze of our system, designed neural network must be run frequently and it causes time complexity overhead. On the other hand there is no method in their proposed approach to solve the report refusal problem and there is no chance for the target agent to introduce his *referee agents* to us and these flaws cause a late convergence problem for neural network or may be in accurate trust estimation.

In the work of Shi et al. [30], a trust model has been introduced to assist decision-making in order to predict the

likely future behavior by analyzing the past behavior. The authors have mostly worked on the environment facilitation, for example the space of possible outcomes has been studied. They believe it is crucial to identify the space of possible outcomes which determines the nature of the associated trust model. The notion of discrete categories is similar to our model in terms of giving more flexibility to the ratings as feedback in order to get more accurate direct interaction estimation. But they have not taken into account the measurements which would unbalance the trust estimation and their decision-makings are solely based on the previous interactions but in our model after a certain amount of time a maintenance is performed to dynamically update the policies adopted.

X. Conclusion

The contribution of this paper is the proposition of a new probabilistic and statistic-based model to secure multi-agent systems in which agents communicate with each other using dialogue games. A framework based upon *trustworthy* and *referee agents* has been presented, as well as several models, of increasing sophistication, for agents to make use of the information communicated to them by other agents they consider trustworthy to determine the trust of further target agents. Our model has the advantage of being computationally efficient and of taking into account four important factors: (1) the trust (from the viewpoint of the evaluator agents) of the *trustworthy agents*; (2) the trust value assigned to target agents according to the point of view of *trustworthy agents*; (3) the number of interactions between *trustworthy agents* and the target agents; and (4) the timely relevance of information transmitted by *trustworthy agents*. Moreover agents perform maintenance in order to evaluate the consulting agents' trust level by comparing the provided information regarding to the target agent's trust level and the actual behavior of the target agent since it has started interaction. The resulting model allows us to produce a comprehensive assessment of the agents' credibility in a software system.

Acknowledgement

The authors would like to thank The Natural Sciences and Engineering Research Council of Canada (NSERC) (application number 341422-07) and Le Fonds Québécois de la Recherche sur la Nature et les Technologies (NATEQ) (application number 2008-NC-119348) for their financial support.

References

[1] A. Abdul-Rahman, and S. Hailes. Supporting trust in virtual communities. In Proc. of the 33rd Hawaii Int. Conf. on System Sciences, IEEE Computer Society Press. 2000.

[2] A. Abdul-Rahman. The PGP trust model. In EDI Forum: The journal of Electronic Commerce, April, 1997.

[3] F. Azzedin and M. Maheswaran. A trust brokering system and its application to resource management in piblic-resource grids. In Proc. of the 18th International Parralel and Sistributed Processing Symposium (IPDPS'04) pp. 22-31, 2004.

[4] J. Bentahar, F. Toni, J-J. Ch. Meyer and J. Labban. A security framework for agent-based systems. In the International Journal of Web Information Systems, Emerald, 2007 (in press).

[5] J. Bentahar and J-J. Ch. Meyer. A new quantitative trust model for negotiating agents using argumentation. In the International Journal of Computer Science and Applications,4(2):1-21, 2007.

[6] J. Bentahar, Z. Maamar, D. Benslimane, and Ph. Thiran. An argumentation framework for communities of web services, IEEE Intelligent Systems, 22(6), 2007.

[7] P.M. Dung, P. Mancarella and F. Toni. Computing ideal sceptical argumentation. Artificial Intelligence, Special Issue on Argumentation in Artificial Intelligence, 2007.

[8] T. Dong-Huynh, N.R. Jennings and N.R. Shadbolt. Certified reputation: How an agent can trust a stranger. In Proceedings of The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1217-1224, Hakodate, Japan, 2006.

[9] 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) pp. 119-154, 2006.

[10] T. Grandison, and M. Sloman. A survey of trust in internet applications. IEEE Communication Surveys & Tutorials, 3(4), 2000.

[11] T.D. Huynh, N.R. Jennings, and N.R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. Journal of Autonomous Agents and Multi-Agent Systems AAMAS, 2006, 119-154.

[12] E.M. Maximilien, and M.P. Singh. Reputation and endorsement for web services. ACM SIGEcom Exchanges, 3(1):24-31, 2002.

[13] D.Ch. Parkes. Iterative combinatorial auctions: achieving economics and computational efficiency. PhD Thesis, University of Pennsylvania, 2001.

[14] S. Parsons, P.J. Gmytrasiewicz, and M.J. Wooldridge (Eds.). Game Theory and Decision Theory in Agent-Based Systems. Springer, 2002.

[15] E. Shakshuki, L. Zhonghai, and G. Jing. An agent-based approach to security service. International Journal of Network and Computer Applications. Elsevier, 28(3): 183-208, 2005.

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

[17] 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.

[18] Dash, R.K, Ramchurn, S.D. and Jennings, N.R. Trust-based mechanism design. Autonomous Agents and Multiagent Systems, AAMAS, pp. 748-755, 2004.

[19] Zhang W. Cpmputational trust model in Online Auctions. Wireless Communications, Networking and Mobile Computing, WiCom 2007, pp.3762 - 3765, 2007.

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

[21] P.B. Velloso, R.P. Laufer, M.B. Duarte and G. Pujolle. HIT: A human-inspired trust model. IFIP International Federation for Information Processing, vol 211, pp. 35-46, 2006.

[22] I.A. Junglas, N.A. Johnson, D.J. Steel, D.Ch. Abraham and P. MacLoughlin. Identifying formation, learning styles and trust in virtual worlds. The DATA BASE for Advances in Information Systems, vol 38, number 4, November 2007.

[23] L. Guiso, P. Sapienza and L. Zingales. Trusting the Stock Market, ECGI - Finance Working Paper No. 170, May 2007.

[24] B. Yu, and M.P. Singh. An evidential model of distributed reputation management. In Proc. of the First Int. Conference on AAMAS. ACM Press, pp. 294-301, 2002.

[25] B. Yu, and M.P. Singh. Detecting deception in reputation management. In Proc. of the 2nd Int. Conference on AAMAS. ACM Press, pp. 73-80, 2003.

[26] B. Yu, and M.P. Singh. Searching social networks. In Proc. of the 2nd Int. Conference on AAMAS, pp. 65-72, 2003.

[27] C. Dellarocas. The digitization of word-of-mouth: promise and challenges of online feedback mechanisms. Management science, 2003, vol 49, no 10, pp. 1407-1424, 2003.

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

[29] W. Song, V.V. Phoha, X. Xu. An adaptive recommendation trust model in multiagent system. Intelligent Agent Technology, 2004. (IAT 2004). Proceedings. IEEE/WIC/ACM, pp. 462- 465, 2004.

[30] J. Shi, G.V. Bochmann and C. Adams. A trust model with statistical foundation. IFIP International Federation for Information Processing, vol 173, pp.145-158, 2005.