

CRM: An Efficient Trust and Reputation Model for Agent Computing

Babak Khosravifar^a, Jamal Bentahar^a, Maziar Gomrokchi^a, and Rafiul Alam^a

^a *Concordia University, Montreal, Canada*

Abstract. In open multi-agent systems, agents engage in interactions to share and exchange information. Due to the fact that these agents are self-interested, they may jeopardize mutual trust by not performing actions as they are expected to do. To this end, different models of trust have been proposed to assess the credibility of peers in the environment. These frameworks fail to consider and analyze the multiple factors impacting the trust. In this paper, we overcome this limit by proposing a comprehensive trust framework as a multi-factor model, which applies a number of measurements to evaluate the trust of interacting agents. First, this framework considers direct interactions among agents, and this part of the framework is called online trust estimation. Furthermore, after a variable interval of time, the actual performance of the evaluated agent is compared against the information provided by some other agents (consulting agents). This comparison in the off-line process leads to both adjusting the credibility of the contributing agents in trust evaluation and improving the system trust evaluation by minimizing the estimation error. What specifically distinguishes this work from the previous proposals in the same domain is its novelty in after-interaction investigation and performance analysis that prove the applicability of the proposed model in distributed multi-agent systems. In this paper, the agent structure and interaction mechanism of the proposed framework are described. A theoretical analysis of trust assessment and the system implementation along with simulations are also discussed. Finally, a comparison of our trust framework with other well-known frameworks from the literature is provided.

Keywords. Trust, reputation, multi-agent systems, agent communication.

1. Introduction

During the past couple of years, agent communication languages and protocols have been of much interest in multi-agent systems, where agents are distributed in large scale networks and interact to collaborate, coordinate and share services and resources [2]. Trust is then essential to make such interactions within open multi-agent systems effective [4,23,25,32]. An agent's trust is a measurement of the agent's possibility to actually do what he agrees to do. Attempting to maintain a trust-based approach, different frameworks have been proposed to represent and assess the trust agents have in one another. The most recent research proposals in trust models for multi-agent systems are as follows: (a) interaction trust, based on the direct interactions of two parties [30,31]; (b) trust based on the type of prior interactions [10,11,22]; (c) witness reputation based on certified (and possibly encrypted) references obtained by the agent to be evaluated after interacting with other agents. These references are then made public to any other agent

who wants to interact with this agent [10,12,17,21,27]; and (d) referenced reputation, based on references from other agents detailing a particular agent's behavior [5,6,10,11].

The proposed frameworks objectively emphasize collecting some parameters that may contribute in the trust assessment procedure. The aim is to collect reliable information leading to an accurate trust assessment process. Since agents might be selfish, receiving fake information by particular agent(s) is always possible. This problem does exist even when a certified reputation [10] is provided by the agent to be evaluated. In this case, the final trust rate would be affected by non-reliable information and eventually the agents' perception of their surrounding environment will not be accurate. Generally, these frameworks are not suitable when the environment changes dynamically because they fail to quickly recognize the recent improvement or degradation of agents' capabilities as in dynamic environments these agents tend to change their goals and behaviors. To overcome this problem, some methods have been proposed to capture the recent changes in the environment [6,18,17]. In these frameworks, a retrospect trust adjustment mechanism is proposed to reconsider the trust evaluations that have been performed in the past to learn how to select better witness agents. Although the mechanism is novel in this domain, its complexity is a considerable issue. Moreover, the applicability of the proposed framework is vague in the sense that the retrospect mechanism does not follow a systematic execution process that enhances the agents' accuracy.

The framework we propose in this paper is built upon a model in which a set of trust meta-data was introduced to define the trust level of contributing agents [18,3,4,5]. The objective of this paper is to overcome the aforementioned limitations by proposing a comprehensive framework called *CRM* (Comprehensive Reputation Model). In this framework, agents interact and rate each other based on previous interactions (either satisfactory or dissatisfactory). The obtained ratings are collected to assess the trustworthiness of a particular agent. To be self-contained, we also consider how agents communicate to exchange ratings. Inter-agent communication is regulated by protocols (shared amongst agents and thus made public) and determined by strategies (internal to agents and thus private) [2]. Using this framework, agents are capable of evaluating the trust level of other agents that are not known (or not very well-known) by collecting some relative information, either from their interaction history or from consulting other agents that can provide their suggestions in the form of ratings. To express the efficiency of the proposed framework, we discuss in more details the performance of the CRM with respect to accuracy, scalability and applicability.

CRM's Accuracy: In general, CRM is based on collecting information before making decisions. The idea of consulting other agents originates from the fact that in social networks, agents assess diverse trust levels for other agents depending on their different experiences of direct and indirect interactions, and thus, an evaluator agent can balance the trust assessment process by considering different factors. In this model, the evaluator agent is referred to as *the trustor agent* and the agent to be evaluated is referred to as *the trustee agent*. In the evaluation process, the trustor may ask some other agents to report on the trustee. These interfering agents are basically divided into two groups: (1) well-known agents by the trustor agent (so-called *trustworthy agents*); and (2) those introduced by the trustee agent (so-called *referee agents*). CRM reaches acceptable accuracy because it collects the information from the agents that are considered the most appropriate sources. The potential aim is on updating the consulting agents to only keep the most accurate ones (i.e. the most trustful). The structure of information update approaches a

stable situation wherein the trustor agent is capable of collecting accurate information from trustworthy set of witness agents and filtering reliable agents.

CRM's Scalability: In general, a trust system is considered scalable when over the population expansion, the complexity does not get affected accordingly. Simply put, system scalability refers to its capability of handling large populations. In the structure that defines the CRM framework, the scalability is considered at best. This is explained by the fact that enlarging the network does not affect the evaluation process according to which the trustor agent always uses a limited number of consulting agents. The framework applies a maintenance process that aims to increase the system accuracy while keeping the size of consulting agents community small. Agents use their historical information and do not initiate a new process of information search upon every request. Therefore, when the number of agents increases, the process of evaluation remains linear with the size of the network. Furthermore, when the network is large enough, the propagation of trust information becomes faster and agents can get more knowledge about the environment rapidly. This makes the agents capable to maintain interactions with new agents.

CRM's Applicability: It is worthy to discuss the applicability of the proposed model. In fact, in distributed multi-agent systems (for example distributed agent-based web services and trading agents in e-commerce settings) the proposed framework is applicable. What makes the proposed model essential in these environments is its sensitivity to obtain accurate information and its capability to survive in dynamic environments. In fact, all the systems that involve multiple components, which require to exchange information need to establish a comprehensive and adaptable trust framework to guarantee the safety of information retrieval.

The formalization and computation of off-line evaluation adjustment made by the trustor after a variable period of direct interactions is the main contribution of this paper. The trustor does this in order to adjust the accuracy of the *consulting agents* (i.e. trustworthy and referee agents). In the off-line process, the suggestions provided by other agents are compared with the actual behavior of the trustee through direct interaction. The trustor will update his beliefs about the consulting agent with respect to the accuracy and usefulness of the provided information through different trust evaluation procedures. By doing this, more accurate ratings about the other agents will be gradually propagated throughout the environment [1], which provides a better trust assessment in the CRM model. In the off-line process, the maintenance mechanism is designed such that it prevents collusion performed between the trustee and referees community. In the off-line process, the consulting agents are evaluated by the trustor agent and because of not being accurate they can get penalized. Therefore, to attract the trustor agent, they need to provide accurate information. We have analyzed the impact of the off-line process from different points of view and compared the system's efficiency with some other models.

The remainder of this paper is organized as follows. In Section 2, we present the specification of agents interaction system together with the trust computing mechanism. Section 3 focuses on the propagation of trust through a social network and defines our framework that combines trustworthy and referee agents as reporters. Afterwards, we describe and discuss the details of computing the trust in our combined framework. In Section 4, we perform the maintenance that typical agent makes after a variable interval of time since the interactions have been initiated. In Section 5, we outline the properties of our model in the experimental environment, present the testbed and compare the simulation results of the CRM model with the results of other well-known trust models in

terms of efficiency in reputation assessments. We also discuss the features of the CRM model and its efficiency, particularly in dynamically changing environments. Section 6 compares our framework with related work, and finally Section 7 concludes the paper.

2. Trust Evaluation Environment

2.1. Interaction System Structure

In this section, we define the communication messages the agents exchange during the trust evaluation process along with the corresponding dialogue game rules.

Definition 1 A communication message is a tuple $\langle \alpha, \beta, Ag_x, Ag_y, M, t \rangle$, where $\alpha \in \{Req, Rep\}$ indicates whether it is a request or reply communication message; $\beta \in \{Inf, Refuse, Not\ Have\}$ represents the type of the message as requesting information in case of initiating the communication (*Inf*), refusing to reveal information (*Refuse*), or not having the information in case of replying to a request message (*Not Have*); agents Ag_x and Ag_y are respectively the sender and receiver of the message; M is the content of the message and finally t is the time at which the message is sent.

Let \mathcal{T}_{Ag_a} be the set of all Ag_a 's trustworthy agents and $\mathcal{T}_{s_{Ag_a}^{Ag_b}} \subseteq \mathcal{T}_{Ag_a}$ be the selected trustworthy agents Ag_a (the trustor) uses to evaluate Ag_b (the trustee). The selection of trustworthy agents is upon need and thus would differ from evaluation to another with respect to the interaction history between the trustor and trustee. In general, ranking and selecting the most trustworthy agents could be applied. The set of selected trustworthy agents is subject to continuous update with respect to environment changes. This issue is discussed in more details later in this paper. To request information, Ag_a uses the communication message $\langle Req, Inf, Ag_a, Ag_{t_1}, Trust(Ag_b), t_0 \rangle$, which means Ag_a at time t_0 sends to the trustworthy agent Ag_{t_1} ($Ag_{t_1} \in \mathcal{T}_{s_{Ag_a}^{Ag_b}}$), a request for information (*Inf*) related to Ag_b 's trust. Consequently Ag_{t_1} replies to the message by one of the following choices:

- 1) $\langle Rep, Inf, Ag_{t_1}, Ag_a, Inf(Ag_b), t_1 \rangle$;
- 2) $\langle Rep, Not\ Have, Ag_{t_1}, Ag_a, *, t_1 \rangle$; or
- 3) $\langle Rep, Refuse, Ag_{t_1}, Ag_a, *, t_1 \rangle$

where $t_1 > t_0$. In the first choice, Ag_{t_1} replies by sending to Ag_a the relative information (trust rating and the number of direct interactions between Ag_{t_1} and Ag_b) about the credibility of Ag_b . In the second choice, Ag_{t_1} informs Ag_a that he does not have any information regarding the credibility of Ag_b (* represents empty message). Finally, in the third choice, Ag_{t_1} refuses to reveal the requested information to Ag_a . There is a chance that Ag_{t_1} replies with *Not Have* reply type in order to hide his refusal of providing information. Such cases are among the situations that Ag_a would consider while adjusting his beliefs about the accuracy of the provided information. Consequently, the non-accurate agents would be penalized in the sense that a trustworthy agent for Ag_a may not be considered in \mathcal{T}_{Ag_a} anymore. These details are out of scope of this paper and here

we only focus on recognizing and thus avoiding the non-accurate agents. The sequence of these request and reply messages represents a dialogue game that we formalize by the following rule, where \Rightarrow is the implication symbol:

$$\begin{aligned} &\langle Req, Inf, Ag_a, Ag_{t1}, Trust(Ag_b), t_0 \rangle \Rightarrow \\ &\quad \langle Rep, Inf, Ag_{t1}, Ag_a, Inf(Ag_b), t_1 \rangle \\ &\quad \vee \langle Rep, Not\ Have, Ag_{t1}, Ag_a, *, t_1 \rangle \\ &\quad \vee \langle Rep, Refuse, Ag_{t1}, Ag_a, *, t_1 \rangle \end{aligned}$$

Meanwhile, Ag_a uses the $\langle Req, Ref, Ag_a, Ag_b, Referee(NUM), t_0 \rangle$ communication message, which means Ag_a at time t_0 sends to Ag_b a request to introduce some referees (Ref). The content message $Referee(NUM)$ indicates the number of referee agents (NUM) that can recommend Ag_b . Ag_b is supposed to introduce the referee agents that support him in the trust evaluation done by Ag_a . Ag_b would rely on his best trustworthy agents in this exercise. Let \mathcal{R}_{Ag_b} be the set of Ag_b 's referee agents. Then, Ag_b after receiving the request communication message, chooses the appropriate referee agents from \mathcal{R}_{Ag_b} . The selected subset, which is introduced to Ag_a at time t_2 ($t_2 > t_0$), is denoted by $\mathcal{R}_{Ag_b}^{Ag_a}$, where $|\mathcal{R}_{Ag_b}^{Ag_a}| = NUM$. This issue is formalized by the dialogue game represented by the following rule:

$$\begin{aligned} &\langle Req, Ref, Ag_a, Ag_b, Referee(NUM), t_0 \rangle \Rightarrow \\ &\quad \langle Rep, Ref, Ag_b, Ag_a, \mathcal{R}_{Ag_b}^{Ag_a}, t_2 \rangle \end{aligned}$$

After obtaining the set of referee agents from Ag_b , Ag_a continues with requesting information from each introduced referee agent at time t_3 . At t_4 ($t_4 > t_3$), the requested referee agent has three possible answers: replying by giving the relative information about the credibility of Ag_b ; replying with no information; or refusing to reveal the information regarding the credibility of Ag_b . Let Ag_{r1} be a selected referee agent ($Ag_{r1} \in \mathcal{R}_{Ag_b}^{Ag_a}$), the following dialogue game rule specifies the exchanged messages:

$$\begin{aligned} &\langle Req, Inf, Ag_a, Ag_{r1}, Trust(Ag_b), t_3 \rangle \Rightarrow \\ &\quad \langle Rep, Inf, Ag_{r1}, Ag_a, Inf(Ag_b), t_4 \rangle \\ &\quad \vee \langle Rep, NotHave, Ag_{r1}, Ag_a, *, t_4 \rangle \\ &\quad \vee \langle Rep, Refuse, Ag_{r1}, Ag_a, *, t_4 \rangle \end{aligned}$$

It is rare that the referee agent does not have information regarding the trust level of Ag_b . This is because the referee has been chosen by Ag_b based on previous direct interactions. But this does not guarantee a positive rating regarding Ag_b 's credibility. The chosen referee agent is in fact facing the trustor Ag_a and since there would be after interaction off-line mechanism, the referee agent would be penalized if provides inaccurate information. Therefore, if the referee agent is not satisfied with Ag_b 's behavior, it is better to retrieve the correct information (bad rating) rather than hiding it (replying "Not Have"). To this end, in case Ag_b has changed his behavior, the referee would rationally retrieve his accurate information to obtain better rate from the trustor agent.

2.2. Trust Computing Mechanism

To compute trust (i.e. credibility) in our model, we first introduce the trust function as follows:

Definition 2 Let \mathcal{A} be a set of agents, \mathcal{D} a set of domains or topics, and T a set of time points. The trust function Tr associates two agents from \mathcal{A} , a domain from \mathcal{D} , and a time point from T with a trust value between 0 and 1:

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

Given some concrete agents Ag_a (the trustor) and Ag_b (the trustee) in \mathcal{A} , some concrete domain D , and a time point t , $Tr(Ag_a, Ag_b, D, t)$ stands for “the trust value associated to the trustee agent Ag_b in domain D at time t by the trustor agent Ag_a ”. To simplify the notation, in the remainder we will omit the domain and time 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 $Tr_{Ag_a}^{Ag_b}$. The reason behind this simplification is that our main contribution in this paper is to equip the agents to efficiently evaluate the trust and get adapted with continuous environment changes. Although the domain is important in trust evaluation (as mainly considered in some trust-based frameworks [26,7]), in this paper we only focus on the adaptation of agents with dynamically changing environment and on how agile the agent is while acting where the trust evaluation is crucial. Furthermore, although the time is omitted from the formulation, it is implicitly represented as the trust function is continuous over T .

To estimate trust, we propose a probabilistic method by investigating the distribution of the random variable X representing the trustworthiness of the trustee agent Ag_b . Let us first consider the case where X takes only two values: 0 (the agent is not trustworthy) or 1 (the agent is trustworthy). Therefore, the variable X follows a *Bernoulli distribution* $\beta(1, p)$ so that $E(X) = p$ where $E(X)$ is the expectation of the variable X and p is the probability that the agent is trustworthy. In this distribution, we have:

$$f(k; p) = p^k(1 - p)^{1-k} \quad \text{for } k \in \{0, 1\}$$

$$E(X) = p; \quad \text{var}(X) = p(1 - p)$$

where $f(k; p)$ is the probability mass function, $\text{var}(X)$ is the variance and p is the probability we are looking for. It is then enough to evaluate the expectation $E(X)$ to find $Tr_{Ag_a}^{Ag_b}$. However, when X is a continuous variable, this expectation is a theoretical mean that should be estimated. To this end, we use the *Central Limit Theorem (CLT)* and the *law of large numbers*. The CLT states that whenever a sample of size n (X_1, \dots, X_n) is taken from any distribution with mean μ , then the sample mean $(X_1 + \dots + X_n)/n$ will be approximately normally distributed with mean μ . As an application of this theorem, the arithmetic mean (average) $(X_1 + \dots + X_n)/n$ approaches a normal distribution of mean μ and standard deviation σ/\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 $\mathcal{T}S_{Ag_a}^{Ag_b}$ and referee agents $\mathcal{R}S_{Ag_b}^{Ag_a}$. These variables follow then the same distribution. They are also independent because 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 agent 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 . In our model defined in depth in the following sections, the mathematical estimation of the expectation $E(X)$ is computed in two steps, on-line and off-line estimation. In the on-line estimation, five main components are considered: direct trust, consulting reports from referee and trustworthy agents, interaction strength, interaction recency, and witness confidence level on the provided information. The off-line estimation, performed after the on-line process, is formulated to modify the trust values of the agents that have provided information in the on-line process. We refer to this process as *maintenance*, which will be addressed in Section 4.

3. On-line Trust Estimation

In this section, we discuss the on-line evaluation process in which the trustor collects some information and combines them to assess the credibility of a trustee. Two approaches can be distinguished in this process. In the former one, the evaluator only relies on what he has from previous interactions with the trustee. In the later, the trustor prefers using the information provided by some other agents to get a more accurate assessment. In the whole process, the direct interaction assessment is combined with the suggested ratings by the consulting agents.

3.1. Direct Trust Evaluation

Agents can compute the trust value of each other using their interaction histories. This would generate real numbers, which fall in the range $[0,1]$ and thus, instead of just integer ratings (scores) 0 and 1, we would have more flexible real ratings representing the satisfaction or dissatisfaction degree of the interaction's outcome. In the general case, agents can evaluate their interactions according to a scale of n types numbered from 1 (the most successful interaction) to n (the least successful interaction), such that the first m interaction types ($m < n$) are successful. Let $NI_{i,Ag_a}^{Ag_b}$ be the number of interactions of type i between Ag_a and Ag_b . Then, $Tr_{Ag_a}^{Ag_b}$ can be computed by Equation 1. This method of direct trust evaluation is similar to the ones proposed in [7,14]. In this Equation, the ratio of the "number of successful outcomes" to the "total number of possible outcomes" is computed, where w_i is the weight associated to the interaction type i to represent its importance and $v_{i,j}$ is the value of the interaction, which is particularly important in transactional settings to avoid two transactions with different values being treated equally. It is worthy to point out that the number of interactions $NI_{i,Ag_a}^{Ag_b}$ is only considered here as a means to evaluate the strength of the connection between the agents Ag_a and Ag_b . In our approach, we do not consider the details of these interactions as it would increase the complexity of the trust evaluation.

$$Tr_{Ag_a}^{Ag_b} = \frac{\sum_{i=1}^m (w_i \times \sum_{j=1}^{NI_{Ag_a}^{Ag_b}} v_{ij})}{\sum_{i=1}^n (w_i \times \sum_{j=1}^{NI_{Ag_a}^{Ag_b}} v_{ij})} \quad (1)$$

In fact, there are two issues in weighting an interaction: 1) the importance of the interaction type (e.g., in some cases *fair* as an interaction's outcome is enough for the interaction to be counted as important, but in other cases, maybe *very good* is mandatory as an outcome type for the interaction to be counted as important enough); and 2) transaction importance (e.g, two transactions of the same type (say *good*) may have different values in terms of their actual entity). Let us consider the following example of two dissatisfactory transactions (e.g., outcome is *bad*) that have been weighted for $w_i = 3$. Basically the value 3 reflects the importance of this kind of transactions (i.e., the weight of *bad* transactions), which could hold different values. v_{ij} is used to represent this value. For example, the first transaction has as value ($v_{i1} = 20000\$$ and the second has as value $v_{i2} = 200\$$). In this example, v_{ij} would reflect the extent to which the damage has been occurred. This idea will protect the model from attacks like reputation squeeze [8] in which one agent would obtain some positive ratings and make a bad interaction that actually makes a large damage.

Another factor should be considered 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. There are some similar approaches in the literature addressing this issue. For example, in [7], the authors discuss the limitations that are used in the freshness of the data to be evaluated. In our model, we assess this factor denoted by $TiR(\Delta t_{Ag_a}^{Ag_b})_{ij}$ by using the function defined in Equation 2 and we do not make the system so sensitive to the past data as it might bring up more confusion to the trustor agent. However, as will be discussed later in this paper, we equip the CRM agent with an off-line mechanism that overcomes this sensitivity. We call this function: the *Timely Relevance* function.

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

The variable λ is application-dependent and $(\Delta t_{Ag_a}^{Ag_b})_{ij}$ is the time difference between the current time (i.e. the time of evaluation of Ag_b by Ag_a) and time at which interaction j of type i took place between these two agents. The intuition behind this formula is to use a function decreasing with the time difference. Consequently, recent information makes the timely relevance coefficient higher. The graph of $TiR(\Delta t_{Ag_a}^{Ag_b})_{ij}$ using different λ values is shown in Figure 1. In some applications, recent interactions are more desirable to be considered when evaluating the trustee. In that case, the trustor uses a higher value for λ . In some other applications, even the old interactions are still valuable sources of information. In that case, the trustor assigns a smaller value to λ . We notice that by calculating $\Delta t_{Ag_a}^{Ag_b}$ as explained above, we do not assume that agents interact continuously in every moment of time; instead, we consider the time of the interaction between two specific agents Ag_a and Ag_b , so that their more recent interactions

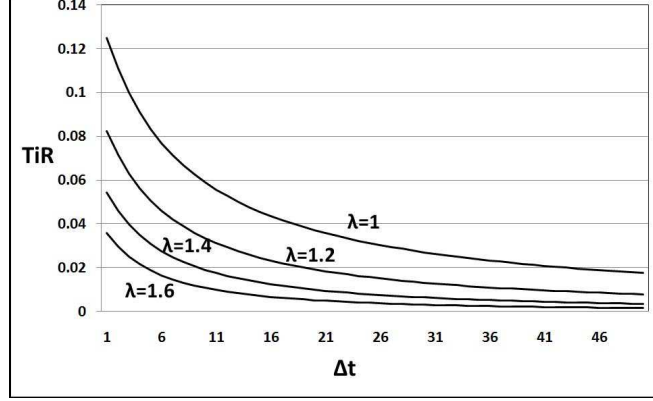


Figure 1. The timely relevance function with respect to different λ values.

are given higher consideration than their old ones. For instance, let t_{10} be the current time, and assume that Ag_a and Ag_b interacted twice before at t_5 and t_8 . The interaction at t_8 is given higher weight than the interaction at t_5 . Considering the involved issues, we recompute the direct trust in Equation 3. In fact, Ag_a rates each previous interaction with Ag_b in terms of its freshness, which privileges recent interactions because they are more valuable sources of information.

$$Tr_{Ag_a}^{Ag_b} = \frac{\sum_{i=1}^m (w_i \sum_{j=1}^{NI_i^{Ag_b}} v_{ij} \times TiR(\Delta t_{Ag_a}^{Ag_b})_{ij})}{\sum_{i=1}^n (w_i \sum_{j=1}^{NI_i^{Ag_b}} v_{ij} \times TiR(\Delta t_{Ag_a}^{Ag_b})_{ij})} \quad (3)$$

3.2. Consulting Reports: Indirect Trust Estimation

The other approach in trust estimation of the trustee consists of collecting some information in terms of suggestions from some other agents. As described before, consulting agents are divided into two groups: (1) trustworthy agents the trustor Ag_a can rely on to request information; and (2) referee agents introduced by the trustee 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 to support Ag_b .

As mentioned before, $\mathcal{T}_{S_{Ag_a}^{Ag_b}}$ is the set of trustworthy agents selected by Ag_a for consultation. 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 he knows. Ag_a consequently asks (some of) the referees to report on the credibility of Ag_b ($\mathcal{R}_{S_{Ag_a}^{Ag_b}}$) and those referees reply according to their past experiences of direct interaction with Ag_b .

Assume there is a particular referee agent Ag_r that Ag_a does not know. In this case, Ag_a does not consider his suggestion about Ag_b , but he saves it anyway in order to compare it with the real behavior Ag_b performs after starting interacting 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 Ag_b 's real behavior and the referee's suggestion.

Let n be the total number of interaction types (see Equation 1) and $NI_{Ag_x}^{Ag_y}$ be the total number of interactions between two agents Ag_x and Ag_y , which is computed by Equation 4:

$$NI_{Ag_x}^{Ag_y} = \min\left(\sum_{i=1}^n NI_{Ag_x}^{Ag_y}, MV\right) \quad (4)$$

In this equation, MV , fixed by the system designer, is the maximum value that $NI_{Ag_x}^{Ag_y}$ can reach after a finite number of interactions. When the number of interactions goes beyond MV , the old interactions are simply not counted, so that only the MV most recent interactions are considered. This restriction makes the model suitable for a large amount of real scenarios where agents have limited resources and computing capabilities. It is worthy to mention that the total number of interactions between Ag_t as a trustworthy agent (resp. Ag_r as a referee agent) and Ag_b , $NI_{Ag_t}^{Ag_b}$ (resp. $NI_{Ag_r}^{Ag_b}$) is an important factor because it promotes information coming from agents knowing more about Ag_b . The agents that had high number of interactions with Ag_b are considered as good sources of information about his trustworthiness in the sense that they are supposed to know Ag_b from relatively longer history of interactions. Considering this factor, Ag_a would penalize the agents with high interactions harder in the maintenance process.

Regarding the importance of the information provided by a consulting agent, we consider another factor, which reflects the confidence (in the range of $[0, 1]$) of the consulting agent on truthfulness of the provided information ($Cf_{Ag_t}^{Ag_b}$ for the typical trustworthy agent and $Cf_{Ag_r}^{Ag_b}$ for the typical referee agent). This factor has a twofold aim. First, the consulting agent would let the trustor agent Ag_a to have a better decision on the extent to which he can take this information into account. Second, the consulting agent would clarify the extent to which he can take the risk on contributing in the trust estimation process initiated by Ag_a . In the simulations, the confidence is randomly generated for each consulting agent using a Gaussian distribution with mean 0.5 and variance 0.2.

The trust equation $Tr_{Ag_a}^{Ag_b}$ we are interested in should take into account the aforementioned relevant factors: (1) the trustworthiness of trustworthy/referee agents according to the trustor Ag_a ($Tr_{Ag_a}^{Ag_t}$ and $Tr_{Ag_a}^{Ag_r}$); (2) the trustee Ag_b 's trustworthiness according to the trustworthy/referee agents ($Tr_{Ag_t}^{Ag_b}$ and $Tr_{Ag_r}^{Ag_b}$); (3) the total number of interactions between these trustworthy/referee agents and Ag_b ($NI_{Ag_t}^{Ag_b}$ and $NI_{Ag_r}^{Ag_b}$), as communicated by Ag_t/Ag_r to Ag_a following the dialogue games previously indicated in Section 2.1; and (4) the confidence of trustworthy/referee agents about the provided information ($Cf_{Ag_t}^{Ag_b}$ and $Cf_{Ag_r}^{Ag_b} \in [0, 1]$). Before defining this equation, let us discuss its desired properties. Some of these properties are inspired by [15].

Property 1 *Assuming that the trustee is known in the system by some agents, $Tr_{Ag_a}^{Ag_b}$ is continuous.*

This property says that at each moment the trustor Ag_a can evaluate the trustee Ag_b . This does not mean that agents are interacting every moment of time, but at every moment, the trustor can get the needed information to assess the trust value of the trustee.

Property 2 Assuming that the trustee is known in the system by some agents, $Tr_{Ag_a}^{Ag_b}$ is strictly monotonically increasing in $Tr_{Ag_t}^{Ag_b}$ and $Tr_{Ag_r}^{Ag_b}$.

This property says that the trust value of the trustee increases if he performs well in his environment. Consequently, agents always have incentives to do better to get their overall trust increased.

Property 3 Assuming that the trustee is known in the system by some agents, $Tr_{Ag_a}^{Ag_b}$ is not monotonically increasing or decreasing in one of the followings: $Tr_{Ag_a}^{Ag_t}$, $NI_{Ag_t}^{Ag_b}$, $Cf_{Ag_t}^{Ag_b}$, $Tr_{Ag_a}^{Ag_r}$, $NI_{Ag_r}^{Ag_b}$, and $Cf_{Ag_r}^{Ag_b}$.

This property says that the trust values of trustworthy agents and trustee are not necessarily correlated. The reason is that some of these agents support the trustee, but some of them do not. The same property holds for referee agents and for the number of interactions and confidence. Thus, for instance, by increasing the number of his interactions with some agents, the trustee cannot guarantee a growth of his trust value, because these agents are probably not supportive.

Property 4 Assuming that the trustee is known in the system by some agents, $Tr_{Ag_a}^{Ag_b}$ is strictly monotonically increasing in one of the followings: $Tr_{Ag_a}^{Ag_t}$, $NI_{Ag_t}^{Ag_b}$, $Cf_{Ag_t}^{Ag_b}$, $Tr_{Ag_a}^{Ag_r}$, $NI_{Ag_r}^{Ag_b}$, and $Cf_{Ag_r}^{Ag_b}$ iff all Ag_t and Ag_r agents support Ag_b .

This property gives the condition on the trustworthy and referee agents, so that increasing their trust value, number of interactions, and confidence will make the trust value of the trustee increasing. The opposite is given by the following property:

Property 5 Assuming that the trustee is known in the system by some agents, $Tr_{Ag_a}^{Ag_b}$ is strictly monotonically decreasing in one of the followings: $Tr_{Ag_a}^{Ag_t}$, $NI_{Ag_t}^{Ag_b}$, $Cf_{Ag_t}^{Ag_b}$, $Tr_{Ag_a}^{Ag_r}$, $NI_{Ag_r}^{Ag_b}$, and $Cf_{Ag_r}^{Ag_b}$ iff all Ag_t and Ag_r agents do not support Ag_b .

Property 6 Let X be the set of all pieces of information that Ag_a uses to assess Ag_b , and Y the set of all pieces of information that Ag_a uses to evaluate another trustee Ag_c , i.e. $X = \{Tr_{Ag_a}^{Ag_t}, Tr_{Ag_t}^{Ag_b}, NI_{Ag_t}^{Ag_b}, Cf_{Ag_t}^{Ag_b} | Ag_t \in \mathcal{T}s_{Ag_a}^{Ag_b} \cup \mathcal{R}s_{Ag_a}^{Ag_b}\}$ and $Y = \{Tr_{Ag_a}^{Ag_{t'}}, Tr_{Ag_{t'}}^{Ag_c}, NI_{Ag_{t'}}^{Ag_c}, Cf_{Ag_{t'}}^{Ag_c} | Ag_{t'} \in \mathcal{T}s_{Ag_a}^{Ag_c} \cup \mathcal{R}s_{Ag_a}^{Ag_c}\}$. Suppose that there is an injective function $f : X \rightarrow Y$ such that for all $x \in X$, $f(x)$ is at least as good for Ag_c as x is good for Ag_b ; then, $Tr_{Ag_a}^{Ag_c}$ is at least as great as $Tr_{Ag_a}^{Ag_b}$.

Let us now define the trust equation $Tr_{Ag_a}^{Ag_b}$ (Equation 5) and then prove it satisfies the aforementioned properties. This equation is composed of two different terms representing the values obtained from two different consulting communities involved in trust evaluation. The functions Ω_T and Ψ_R are defined as the combination of the trust values estimated by the trustworthy and referee agents together with their related trustworthiness from Ag_a 's point of view, timely relevance, confidence and number of interactions between the trustworthy and referee agents and the trustee Ag_b .

$$Tr_{Ag_a}^{Ag_b} = \frac{\Omega_T(\mathcal{T}s_{Ag_a}^{Ag_b}) + \Psi_R(\mathcal{R}s_{Ag_a}^{Ag_b})}{\Omega'_T(\mathcal{T}s_{Ag_a}^{Ag_b}) + \Psi'_R(\mathcal{R}s_{Ag_a}^{Ag_b})} \quad (5)$$

where

$$\Omega_T(\mathcal{T}s_{Ag_a}^{Ag_b}) = \sum_{Ag_t \in \mathcal{T}s_{Ag_a}^{Ag_b}} Tr_{Ag_a}^{Ag_t} \times Tr_{Ag_t}^{Ag_b} \times NI_{Ag_t}^{Ag_b} \times Cf_{Ag_t}^{Ag_b}$$

$$\Omega'_T(\mathcal{T}s_{Ag_a}^{Ag_b}) = \sum_{Ag_t \in \mathcal{T}s_{Ag_a}^{Ag_b}} Tr_{Ag_a}^{Ag_t} \times NI_{Ag_t}^{Ag_b} \times Cf_{Ag_t}^{Ag_b}$$

$$\Psi_R(\mathcal{R}s_{Ag_a}^{Ag_b}) = \sum_{Ag_r \in \mathcal{R}s_{Ag_a}^{Ag_b}} Tr_{Ag_a}^{Ag_r} \times Tr_{Ag_r}^{Ag_b} \times NI_{Ag_r}^{Ag_b} \times Cf_{Ag_r}^{Ag_b}$$

$$\Psi'_R(\mathcal{R}s_{Ag_a}^{Ag_b}) = \sum_{Ag_r \in \mathcal{R}s_{Ag_a}^{Ag_b}} Tr_{Ag_a}^{Ag_r} \times NI_{Ag_r}^{Ag_b} \times Cf_{Ag_r}^{Ag_b}$$

We notice that $Tr_{Ag_a}^{Ag_t} \neq 0 \forall Ag_t \in \mathcal{T}s_{Ag_a}^{Ag_b}$ and Ag_b is known for at least one Ag_t , which means $NI_{Ag_t}^{Ag_b}, Cf_{Ag_t}^{Ag_b} \neq 0$, so $\Omega'_T(\mathcal{T}s_{Ag_a}^{Ag_b}) \neq 0$.

We now show that Equation 5 satisfies Properties 1 to 6. To simplify the notation, we will omit the arguments of the functions $\Omega_T, \Omega'_T, \Psi_R,$ and Ψ'_R . $Tr_{Ag_a}^{Ag_t}$ and $Cf_{Ag_t}^{Ag_b}$ are non-zero continuous functions on time, and $Tr_{Ag_t}^{Ag_b}$ is continuous on time, so by considering $NI_{Ag_t}^{Ag_b}$ as a coefficient for $Tr_{Ag_t}^{Ag_b}$ for each Ag_t , we conclude that Ω_T and Ω'_T are non-zero continuous functions. Similarly, Ψ_R and Ψ'_R are continuous, so the trust function is continuous. To show that Property 2 is satisfied, we need to prove that the partial derivative of the trust function with respect to $Tr_{Ag_t}^{Ag_b}$ is greater than zero, and the same thing with respect to $Tr_{Ag_r}^{Ag_b}$. To simplify the proof, but without loss of generality, let us consider a specific agent Ag_{t1} , then the same procedure can be applied to all other Ag_t agents. We have:

$$\frac{\partial Tr_{Ag_a}^{Ag_b}}{\partial Tr_{Ag_{t1}}^{Ag_b}} = \frac{Tr_{Ag_a}^{Ag_{t1}} \cdot NI_{Ag_{t1}}^{Ag_b} \cdot Cf_{Ag_{t1}}^{Ag_b}}{\Omega'_T + \Psi'_R} > 0$$

The same proof can be used for a specific referee agent Ag_{r1} , thus the satisfaction of Property 2. To show that Property 3 is satisfied, we need to show that the partial derivative of the trust function with respect to the factors mentioned in this property is not always positive and not always negative. Here we only show the proof for the case $Tr_{Ag_a}^{Ag_t}$ and the same proof can be used for $Cf_{Ag_t}^{Ag_b} \times NI_{Ag_t}^{Ag_b}$ (the number of interactions is considered as a coefficient) and for the other factors. As we did for Property 2, we consider a specific trustworthy agent Ag_{t1} and the generalization follows. We have:

$$\frac{\partial Tr_{Ag_a}^{Ag_b}}{\partial Tr_{Ag_{t1}}^{Ag_b}} = \frac{(Tr_{Ag_{t1}}^{Ag_b} \cdot NI_{Ag_{t1}}^{Ag_b} \cdot Cf_{Ag_{t1}}^{Ag_b}) \cdot (\Omega'_T + \Psi'_R) - (\Omega_T + \Psi_R) \cdot (NI_{Ag_{t1}}^{Ag_b} \cdot Cf_{Ag_{t1}}^{Ag_b})}{(\Omega'_T + \Psi'_R)^2}$$

The sign of this partial derivative depends then on the sign of the numerator, which could be positive or negative. Thus, to prove that Properties 4 and 5 are satisfied, we only need to analyze when the numerator is strictly positive, and when it is strictly negative. We have:

$$\begin{aligned}
& (Tr_{Ag_{t1}}^{Ag_b} \cdot NI_{Ag_{t1}}^{Ag_b} \cdot Cf_{Ag_{t1}}^{Ag_b}) \cdot (\Omega'_T + \Psi'_R) - (\Omega_T + \Psi_R) \cdot (NI_{Ag_{t1}}^{Ag_b} \cdot Cf_{Ag_{t1}}^{Ag_b}) > 0 \\
& \text{iff } Tr_{Ag_{t1}}^{Ag_b} > \frac{\Omega_T + \Psi_R}{\Omega'_T + \Psi'_R} \\
& \text{iff } Tr_{Ag_{t1}}^{Ag_b} > Tr_{Ag_a}^{Ag_b}
\end{aligned}$$

Thus, the partial derivative is strictly positive iff Ag_{t1} is supportive (Property 4), and it is strictly negative iff Ag_{t1} is not supportive (Property 5). If it is equal to zero, the function is simply constant. Finally, to prove that Property 6 is satisfied, we define the injective function f as follows: $f(Tr_{Ag_a}^{Ag_t}) = Tr_{Ag_a}^{Ag_t}$; $f(Tr_{Ag_t}^{Ag_b}) = Tr_{Ag_t}^{Ag_b}$; $f(NI_{Ag_t}^{Ag_b}) = NI_{Ag_t}^{Ag_b}$; and $f(Cf_{Ag_t}^{Ag_b}) = Cf_{Ag_t}^{Ag_b}$. So, for all $x \in X$, $f(x)$ is at least as good for Ag_c as x is good for Ag_b iff $f(x) \geq x$. Consequently, from Property 4, we obtain $Tr_{Ag_a}^{Ag_c} \geq Tr_{Ag_a}^{Ag_b}$, which is the result we want to prove.

Equation 5 is used by the initial trustor Ag_a to evaluate the trustee Ag_b where each consulting agent is supposed to forward his own estimation (together with his confidence level) for this trustee. Following the ideology that Ag_a could, to a certain extent, rely on his own history of interactions with Ag_b (direct trust evaluation approach) and partially use the second approach (indirect approaches), Ag_a gives a 100% trustworthy rate to his history and considers himself as a member of his trustworthy community. This aggregation method takes into account the proportional relevance of each approach, rather than treating the two approaches separately. Basically, the contribution percentage of each approach in the final evaluation of $Tr_{Ag_a}^{Ag_b}$ is defined regarding how informative the history is in terms of the number of direct interactions between Ag_a and Ag_b and their time recency. Therefore, consulting other agents is considered with less importance if the history represents a lower uncertainty. Doing so, the indirect evaluation approach is combined with the direct approach to end up with an accurate trust estimation of the trustor Ag_a for the trustee Ag_b . To be more precise, we aim to analyze the quality of the interactions of the trustee considering what is expected (final trust evaluation $Tr_{Ag_a}^{Ag_b}$) and what is actually performed. To this end, we have a retrospect trust evaluation, which is represented in Section 4.

4. Off-line Trust Estimation

To avoid exposing the reputation framework to dishonest ratings, two types of agents should be considered: (a) bad mouthers: agents who exaggerate by giving negative ratings; and (b) ballot stuffers: agents who exaggerate by giving positive ratings. Minimizing the effects caused by these two types of consulting agents is an important aspect in trust evaluation. Although the ratio of relationship strength can be certainly inserted as a measure of trust to increase the accuracy of referee agent's credibility, this technique is not generic as it depends on how this relationship strength is represented and measured.

To tackle this problem, we propose other parameters. First, we consider the number and time recency of interactions as factors that reflect the trustor's expectation of receiving accurate information. Second, we consider the confidence level provided by consulting agents as a means to enable the trustor to update his friend list. To this end, we split the off-line trust estimation into two parts: Off-line Interaction Inspection and Maintenance.

4.1. Off-line Interaction Inspection

After each interaction, the trustor Ag_a performs an off-line interaction inspection process regarding each of the consulting agents role in the trust evaluation process. In this procedure, Ag_a considers the rate provided by the consulting agent $Ag_c \in (\mathcal{T}s_{Ag_a}^{Ag_b} \cup \mathcal{R}s_{Ag_a}^{Ag_b})$, number and recency of interactions done with the trustee agent Ag_b , and the confidence $Cf_{Ag_c}^{Ag_b}$. The objective of this process is to assign a flag (useful/useless) for each involved consulting agent.

Since the off-line interaction inspection is a process performed after the interaction for a variable interval of time $[t_1, t_2]$, Ag_a has a self opinion about the credibility of Ag_b . Therefore, we refer to $OTR_{Ag_a}^{Ag_b}$ as the actual credibility observed by the trustor Ag_a at t_2 . This value is compared to the rate provided by each consulting agent during $[t_1, t_2]$. Figure 2 is the off-line interaction inspection algorithm that takes the observed trust value ($OTR_{Ag_a}^{Ag_b}$), provided rate by each consulting agent Ag_c ($Tr_{Ag_c}^{Ag_b}$), the corresponding number of interactions ($NI_{Ag_c}^{Ag_b}$) and the provided information time recency ($TiR_{Ag_c}^{Ag_b}$) as input. This algorithm provides an array (called *flag*) of binary numbers about the usefulness of the information provided by each involved consulting agent.

In this algorithm, first the average of the differences between the provided trust and observed one of all the consulting agents is evaluated. The rational behind this is explained by the fact that the public opinion affects the threshold of the accuracy of credibility rating. This means if the average difference is relatively high, the trustor agent Ag_a would doubt that the trustee agent Ag_b is a consistent reliable agent, otherwise the public opinion about this agent would not achieve that divergency. Once the average difference is obtained, the consulting agents are checked one by one to be tagged either as useful or useless. The agents who provided relatively accurate ratings with an acceptable confidence level $Cf_{Ag_c}^{Ag_b} > \nu$ (ν is application-dependant and in the simulations we assume that $\nu = 0.5$) are not all tagged as useful. They are all good except the ones who do not have high number of interactions or time relevance (strong connection or holding fresh information). This is due to the fact that in credibility assessment, the ratings that are submitted at random (by chance) could not be considered as a means to evaluate the truthfulness of a consulting agent. In this algorithm, the number of interactions and time relevance of the consulting agents are compared with the ones about the trustor and trustee agents' connection. To this end, there is higher priority assigned to consulting agents that hold stronger relationship. This partition of consulting agents based on useful and useless flags is an operational way of obtaining the partition of agents as reliable and doubtful as proposed in the TRSIM framework [6].

4.2. Maintenance

The maintenance procedure is a process initiated at different intervals of time to update the information that the trustor agent Ag_a has about his surrounding environment (i.e.

```

function interactionInspector( $Ag_a, Ag_b, \mathcal{T}s_{Ag_a}^{Ag_b}, \mathcal{R}s_{Ag_a}^{Ag_b}, Tr_{Ag_a}^{Ag_b}, OTr_{Ag_a}^{Ag_b}$ ); flag
     $D = 0$ ;     $\bar{D} = 0$ 

    for all  $Ag_c \in \mathcal{T}s_{Ag_a}^{Ag_b} \cup \mathcal{R}s_{Ag_a}^{Ag_b}$ 
         $D_{Ag_c} = |Tr_{Ag_c}^{Ag_b} - OTr_{Ag_c}^{Ag_b}|$ ;
         $D = D + D_{Ag_c}$ ;
    end for all

     $\bar{D} = \frac{D}{|\mathcal{T}s_{Ag_a}^{Ag_b} \cup \mathcal{R}s_{Ag_a}^{Ag_b}|}$ ;

    for all  $Ag_c \in \mathcal{T}s_{Ag_a}^{Ag_b} \cup \mathcal{R}s_{Ag_a}^{Ag_b}$ 
        if  $D_{Ag_c} < \bar{D}$  and  $Cf_{Ag_c}^{Ag_b} > \nu$ 
            if  $NI_{Ag_c}^{Ag_b} > NI_{Ag_a}^{Ag_b}$     flag[ $Ag_c$ ]=useful
            else if  $TiR_{Ag_c}^{Ag_b} > TiR_{Ag_a}^{Ag_b}$     flag[ $Ag_c$ ]=useful
            else    flag[ $Ag_c$ ]=useless
        else    flag[ $Ag_c$ ]=useless
    end for all
end function

```

Figure 2. After interaction inspection algorithm for assigning usefulness flags to each involved consulting agent

about consulting agents). Therefore, the maintenance is not a continuous process, as this choice is computationally expensive and not effective because sometimes, there is no need to update information. In addition, this process, devoted to the trustor agents, does not need an update of all the factors used in Equation 5. Thus, there is no need to update the confidence $Cf_{Ag_t}^{Ag_b}$ and $Cf_{Ag_r}^{Ag_b}$ since it is related to the consulting agents, not the trustor. Before performing this process, there are two questions that have to be addressed: (1) when does the trustor agent need to initiate the maintenance?; and (2) which agents have to be cleared in the maintenance? In the rest of this section, we answer these two questions in more details.

(1) When to initiate the maintenance procedure?

There are three answers for this question:

- **Bad Performance:** This is the case where the performance of evaluating agents decreases below a predefined threshold ($1 - Tr_{Ag_a}$). Tr_{Ag_a} is in fact the reputation value that Ag_a has in the system as estimated by himself using his interactions with other agents. This value does not have to be known publicly as it is used by Ag_a to perform a sort of internal maintenance. In the case of bad performance, the trustor agent realizes that his performance $P_t(Ag_a)$ in trust evaluations (regarding time t) is decreasing in almost a continuous manner. The performance of evaluation is always calculated since the most recent maintenance and is aggregated (in average) over the interval of time since this last maintenance. Let $S(t)$ be the set of trustee agents that Ag_a tries to evaluate since the last maintenance. Equation 6 computes the current performance ($P_t(Ag_a)$) of the trustor agent Ag_a

at time (t) since the last maintenance.

$$P_t(Ag_a) = \frac{\sum_{Ag_b \in S(t)} |Tr_{Ag_a}^{Ag_b} - OTR_{Ag_a}^{Ag_b}|}{|S(t)|} \quad (6)$$

The trustee agent Ag_b is selected from the set $S(t)$. In this process, if $P_t(Ag_a) > 1 - Tr_{Ag_a}$, the trustor agent Ag_a applies a new maintenance process.

- **Huge Difference:** This is the case where Ag_a is disappointed with a noticeable low quality trust evaluation that is recently done. In this case, Ag_a realizes that the provided information is not satisfactory to the extent to which Ag_a can rely on to continue his upcoming evaluations. Therefore if the following inequality holds, the trustor agent will decide to run a new maintenance process as an exceptional case to update his belief set. The value η depends on how picky the evaluator is. In our simulations we assume $\eta = 0.5$. For instance, picky agents can consider $0.2 < \eta \leq 0.5$ and very picky agents can consider $0 < \eta \leq 0.2$. In fact, to enable an equity treatment for all the trustee agents, the threshold should be the same for all these agents, which justifies the use of a fixed value instead of a probabilistic approach.

$$|Tr_{Ag_a}^{Ag_b} - OTR_{Ag_a}^{Ag_b}| > \eta$$

- **After Certain variable interval of Time:** If during the evaluation process there was no problem that caused initiation of a maintenance procedure, the off-line trust estimation system would run, after a certain interval of time, the maintenance process to update the belief set. This would help to have a better adaptation in case of rapid changes in surrounding agents' behavior.

(2) Which agents have to be cleared in the maintenance?

In the maintenance process, Ag_a selects some agents so that applying the maintenance on them would enhance his adaptation with the surrounding environment. In fact, if in the process of trust evaluation, since the most recent maintenance, Ag_a 's belief set has not been changed, Ag_a would consult with the same set of trustworthy agents. All these agents are then included in the maintenance process. Besides these agents, some referee agents probably were involved in some trust evaluations. Ag_a selects the referees that did provide the asked information (regarding different trustee agents) with relatively high confidence ($> \nu$, which is set by Ag_a). The reason behind this is that the process of indirect trust evaluation is in fact a twofold aimed process. Besides obtaining accurate information, Ag_a would like to get to know new agents and to better know the previously known agents. In this case, the truthfulness of the agents regarding the provided information could be considered as a means to get their credibilities updated. However, Ag_a would not consider any referee agent. In the maintenance process, Ag_a only considers the referee agents with high confidence on their provided information. This would let Ag_a apply the update in a more reliable manner.

Let $UF_{Ag_m}^{t_1, t_2}$ and $UL_{Ag_m}^{t_1, t_2}$ be the set of useful and useless flags associated with a trustworthy or referee agent Ag_m from his interactions during the interval $[t_1, t_2]$ as computed by the algorithm given in Figure 2. Equation 7 gives the rate illustrating the performance of Ag_m at time t_2 considering t_1 as a point of reference. This performance

is computed in terms of the number of useful and useless flags during $[t_1, t_2]$, where -1 reflects the worst performance (all the flags are useless), 0 the average performance (the numbers of useful and useless flags are equal), and 1 the best performance (all the flags are useful). This rate is used to update the trust value of Ag_m at t_2 ($Tr_{Ag_a}^{Ag_m}(t_2)$) as illustrated by Equation 8. This update satisfies the properties that 1) if the performance is average ($\alpha_{Ag_m}(t_2) = 0$), then the trust is constant ($Tr_{Ag_a}^{Ag_m}(t_2) = Tr_{Ag_a}^{Ag_m}(t_1)$); 2) if the performance is the worst, then $Tr_{Ag_a}^{Ag_m}(t_2) = 0$; and 3) if the performance is very good, then $Tr_{Ag_a}^{Ag_m}(t_2)$ can achieve 1 depending on the value of ($Tr_{Ag_a}^{Ag_m}(t_1)$).

$$\alpha_{Ag_m}(t_2) = \frac{|UF_{Ag_m}^{t_1, t_2}| - |UL_{Ag_m}^{t_1, t_2}|}{|UF_{Ag_m}^{t_1, t_2}| + |UL_{Ag_m}^{t_1, t_2}|} \quad (7)$$

$$Tr_{Ag_a}^{Ag_m}(t_2) = \begin{cases} 1 & \text{if } Tr_{Ag_a}^{Ag_m}(t_1) \cdot (1 + \alpha_{Ag_m}(t_2)) > 1 \\ Tr_{Ag_a}^{Ag_m}(t_1) \cdot (1 + \alpha_{Ag_m}(t_2)) & \text{if } 0 < Tr_{Ag_a}^{Ag_m}(t_1) \cdot (1 + \alpha_{Ag_m}(t_2)) \leq 1 \\ \alpha_{Ag_m}(t_2) & \text{if } Tr_{Ag_a}^{Ag_m}(t_1) \cdot (1 + \alpha_{Ag_m}(t_2)) = 0 \text{ and } \alpha_{Ag_m}(t_2) > 0 \\ 0 & \text{else} \end{cases} \quad (8)$$

Figure 3 shows the pseudo-code of the maintenance process that computes $Tr_{Ag_a}^{Ag_m}(t_2)$. Ag_a initiates this process with respect to any of the three discussed answers to question 1. In this pseudo-code, \mathcal{M}_{Ag_a} is the set of agents that are going to be selected for the maintenance and as mentioned before, all the trustworthy agents \mathcal{T}_{Ag_a} are included. For all the interactions since the latest maintenance, the trustee is considered. For all the referees of the trustee in question, the selected ones are those who showed high confidence. Finally, with respect to their flags (useful $+UF$ and useless $-UL$), their update rates (α_{Ag_m}) are computed as shown in Equation 7 (with a notational simplification). Then the updated trust value is computed as illustrated in Equation 8.

Since between the variable maintenance periods the trustworthy agents of a particular trustor agent Ag_a are the same, there is a fixed number of agents that are involved in the maintenance process. Moreover, there are some referee agents that are considered in this process and might be different with respect to different trustee agents. Because the number of involved agents in such a process is not high, the corresponding computations regarding their trust value update is negligible in the off-line trust estimation mechanism. Besides this, the trustor agent Ag_a takes the advantage of updating his trust values with respect to the referee agents that might not have high number of interactions. Furthermore, the maintenance algorithm is linear with both the number of agents and the number of interactions (i.e. $O(|\mathcal{T}_{Ag_a}| + \prod_{Ag_e \in S(t)} |\mathcal{R}_{S_{Ag_e}^{Ag_a}}|)$) where $|S(t)|$ is the number of interactions with different trustee agents (say Ag_e as a particular trustee agent), $|\mathcal{T}_{Ag_a}|$ is the number of trustworthy agents, and $|\mathcal{R}_{S_{Ag_e}^{Ag_a}}|$ is the number of referee agents for a given trustee agent Ag_e . We notice that we need to compute all the interactions with referee agents even if some of them are common to different trustee agents, which justifies the product over those trustee agents $S(t)$. The linear complexity of the proposed maintenance process makes it computationally efficient.

```

function maintenance( $Ag_a, S(t)$ )
   $\mathcal{M}_{Ag_a} := \mathcal{T}_{Ag_a}$ ;

  for all  $Ag_e \in S(t)$ 
    for all  $Ag_c \in \mathcal{R}_{Ag_a}^{Ag_e}$ 
      if  $Cf_{Ag_c}^{Ag_e} > \nu$ 
         $\mathcal{M}_{Ag_a} := \mathcal{M}_{Ag_a} \cup \{Ag_c\}$ 
      end for all
    end for all

  for all  $Ag_m \in \mathcal{M}_{Ag_a}$ 
    consider all interactions since last maintenance
     $\alpha_{Ag_m} = \frac{+UF-UL}{+UF+UL}$ 
    if  $Tr_{Ag_a}^{Ag_m} \neq 0$ 
       $X := Tr_{Ag_a}^{Ag_m} \times (1 + \alpha_{Ag_m})$ 
      if  $X > 1$ 
         $Tr_{Ag_a}^{Ag_m} := 1$ 
      else  $Tr_{Ag_a}^{Ag_m} := X$ 
      end if
    else
      if  $\alpha_{Ag_m} \geq 0$ 
         $Tr_{Ag_a}^{Ag_m} := \alpha_{Ag_m}$ 
      else  $Tr_{Ag_a}^{Ag_m} := 0$ 
      end if
    end if
  end for all
end function

```

Figure 3. The maintenance algorithm for updating trust rating performed by the trustor Ag_a

5. Analysis and Experimental Simulation

5.1. Implemented Testbed

In this section, we assess the CRM model efficiency and describe the implementation of the testbed¹. We also compare our model with five well known models as benchmarks: FIRE [10,11], Referral [33,34], SPORAS [35], Travos [27] and BRS [12]. All these models are explained in details and discussed in the related work section (Section 6). The testbed environment (represented in Table 1) is populated with 200 agents categorized by two agent types: (1) service provider agents that are supposed to provide services (for simplicity, we assume that only one type of service is provided and therefore consumed); and (2) service consumer agents (equipped with the different trust models) that are looking for service providers to interact with and consume the provided service. As in FIRE and Travos, in the rest of this paper we use the *gained utility* as a measurement for the quality of obtained service (QoS) in terms of satisfaction, response time, price, etc. Thus,

¹The code is open source and can be downloaded from:
<http://users.encs.concordia.ca/~bentahar/CRM/CRM.zip>

Table 1. Testbed environment

Service Provider Agents (S.P.)	S.P. Agent Type	Density in the S.P. Community	Provided Utility at Each RUN		Radius of Activity
			Range	Standard Deviation	
	Good	15.0%] +5, +10]	1.0	25
	Ordinary	30.0%] -5, +5]	2.0	28
	Bad	15.0%] -10, -5]	2.0	25
	Fickle	40.0%] -10, +10]	-	30
Service Consumer Agents (S.C.) Group1	S.C. Agent Type	Density in the S.C. Community	Number of Joining Agents at Each RUN		Radius of Activity
	CRM	25.0%	6		35
	FIRE	25.0%	6		35
	REFERRAL	25.0%	6		35
	SPORAS	25.0%	6		35
Service Consumer Agents (S.C.) Group2	S.C. Agent Type	Density in the S.C. Community	Number of Joining Agents at Each RUN		Radius of Activity
	CRM	33.3%	10		35
	Travos	33.3%	10		35
	BRS	33.3%	10		35

the gained utility depends on the performance of service provider. We consider two service consumer groups to be compared with our CRM model: (1) *group1* (FIRE, Referral and SPORAS); and (2) *group2* (Travos and BRS). The criterion used in this separation is the degree of sensitivity of the models to the environment and changes of behavior of the service providers. Group1 does not consider the continuous change of agents behaviors. The agents in this group tend to accurately maintain the trust process rather than putting effort on updating trust regarding the environment changes. Group2 takes action in response to such changes more rapidly. Generally, service providers are different and thus provide diverse range of service qualities. Furthermore, the consumer agents using these services obtain different gained utilities. Based on the parameters and strategies the other frameworks use, we have implemented agents that follow these strategies and compared the CRM agents with.

The simulation consists of a number of consequent runs in which agents are activated and build their private knowledge, keep interacting with one another, gain utility and enhance their overall knowledge about the environment. The more an agent knows the environment, the better he can choose service providers and thus, the more utility he gains. Agents are free to ask others for their beliefs about the service provider to be selected. Finally, each agent requests the service from the most trustworthy and reliable provider according to him. Table 1 represents the four types of service providers we consider in our simulation: *good*, *ordinary*, *bad* and *fickle*. The first three provide services according to the assigned mean value of quality with a small range of deviation. However, fickle providers are more flexible as their range of quality covers all possible outcomes. To put the system in a more tight situation, we use a high number of fickle agents. The radius of activity is used to measure how much the provider agent is known in the network to get selected. Each agent has his own network, which is considered in our model as a circle around this agent, which means this agent is his center. Thus, high value of this radius means the agent has a large network. This is the reason why there is a high value assigned for fickle and ordinary providers because we want them to have a high chance to be selected.

Since the major difference between the considered models is the trust mechanism they employ for credibility assessment, the utility gained by each model is considered as its efficiency in selecting reliable service providers. Doing so, we compare CRM with other models in two perspectives, honest (Section 5.2) and biased (Section 5.3) environments. In honest environments, agents are supposed honest in the sense they truthfully reveal their beliefs. However, in biased environments, agents can reveal inaccurate information. Comparison is done first between CRM, FIRE [10] (a successful trust model with high performance), SPORAS [35] (a centralized approach), and Referral [33] (following the concept of reference in an honest environment). Travos [27] and BRS [12] are the two other models that we compared CRM with in terms of how they survive in biased environment where agents constantly change their behaviors. Like CRM, Travos and BRS are designed to take actions while agents are not fully trustworthy. These models differ from CRM in the trust assessment mechanism and analysis they perform in order to choose the best possible provider. In such an environment where agents have an intermittent attitude, a successful trust model is the one that gets adapted with new situations.

5.2. Honest Environment

Figure 4 depicts the overall comparison of different models. The testbed consists of a number of runs represented as the horizontal axis, and the ranking mean value for the utility gained of each group is represented in the vertical axis. Over the runs, each service consumer uses a particular model to find the most trustworthy service provider and thus gain the most utility. First, the mean value of the gained utility by agents using the same trust model is computed. Then, the mean values obtained from different trust models are compared with each other using two sample *t-test* with 95% of confidence level to show the overall outperforming of CRM and FIRE compared to the other two models.

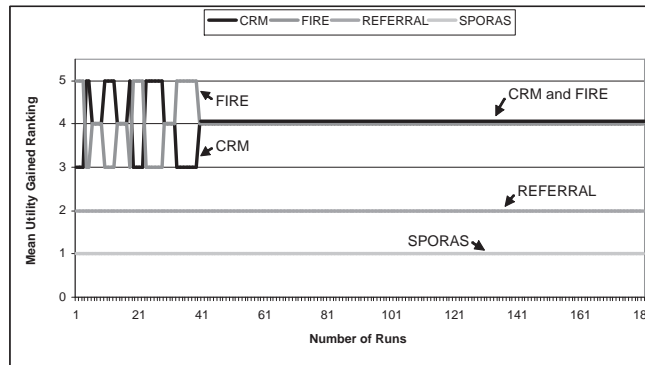


Figure 4. Comparison of CRM with FIRE, Referral and Sporas in terms of mean utility gained at each run in an honest environment

As shown in Figure 4, SPORAS's performance is poor. The reason is that SPORAS model evaluates the trust based on very recent interactions of each agent. Moreover, in this model, the credibility of highly interacted agents undergo a minor change compared to the ones with low number of interactions. Since SPORAS (generally used as benchmark in the literature) is a centralized model, it suffers from inconsistency of the trust

values associated to agents while they register upon entrance in the system. Thus, this model would not perform well in situations when the good service providers are new to the system and remain unknown for a long time. Moreover, we still observe the problem of fake advertising to the central agent to get more benefit. Therefore, SPORAS performance in selecting the best service providers is poor. Referral model agents directly consider how to place trust in others and emphasize the key properties that affect the trust assessment. However, they do not restrict the suggestions of other agents, which lead them to assess the credibility of an unknown or partially known service provider. This may impact the selection of good providers from the beginning of simulation. FIRE agents [10] regulate the problem of collecting the required information by the evaluator to assess the trust of his partner. In addition, they apply certified reputation introduced by the trustee agent. As shown in Figure 4, the commutative utility gained over the 500 elapsed runs by the FIRE and CRM agents are culminated to be the highest as both methods select good service providers, and therefore gain the highest possible utility (for space reasons, only the first 180 runs are shown in this figure). In the first 40 runs, the CRM and FIRE agents obtain different rankings in the set $\{3, 4, 5\}$, which reflects their different capabilities in getting to know the surrounding environment. In this environment, the agents are considered honest in revealing their beliefs. In the next section, we carry on comparison in the biased environment in which agents would untruthfully reveal their beliefs. As a result, the trustor can get confused in the trust assessment. Objectively, we discuss how the CRM agents cope with such a problem.

5.3. *Biased Environment*

Being more realistic, we exposed the same models in a very biased environment in which agents, serving some certain goals, may reveal much less accurate information. Each agent employs his corresponding trust model to accumulate the utility gained through interactions. In general, the agents with more adaptable trust framework would be able to have more efficient performance and thus, obtain higher utility from the environment.

To prove the applicability of the proposed framework, we discuss the features allowing the CRM model to perform higher over the FIRE, Travos and BRS models in terms of efficiency. This discussion considers two perspectives. The former is in terms of balancing the trust assessment process by considering different involved agents. This comparison is done between FIRE and CRM, which is also highlighted with a detailed scenario. The latter discussion focuses on how agents are sensitive to the environment inconsistency and how it would be possible to gain more from diverse types of service providers. The CRM model is compared with the Travos and BRS models to show how these dynamic models act in an extensive intermittent environment.

FIRE is a successful trust-certified reputation model, which addresses the problem of lack of direct history. Agents evaluate the trust of other agents as decentralized services. However, the FIRE agents do not quickly recognize the agents that have got the good ratings and performed bad either in terms of inaccurate ratings provided for some others or the bad obtained utility. The CRM agents are equipped with a maintenance mechanism, which enables them to quickly recognize change of behavior of others and respectively adjust their beliefs regarding the trust of some particular consulting agents. This mechanism is also effective in recognizing collusion behavior, by which agents intentionally reveal inaccurate information, aiming to gain more benefit at the end. This

change of behavior should be recognized and the benefit of other agents should get adjusted. This process helps in quickly recognizing the fickle agents that may provide any quality of service.

Figure 5 shows a graph plotting fickle selection percentage versus number of runs. The graph highlights the difference of having and missing the maintenance regarding the behavior of the CRM and FIRE agents. In the first 80 runs, we observe that the CRM agents are reducing the selection of fickle agents as the time goes on. This is because the CRM agents perform maintenance on the behavior of the fickle agents that provide a bad utility after the interaction, which leads to less selection afterwards. The performance of the FIRE agents remain almost the same as they do not recognize the fluctuated behavior of the fickle agents. The picks of the CRM graph (P_1 and P_2) are simply because of a selection of few number of the CRM agents at each run, and therefore, the maintenance they perform generally has low effect on the consequent run until they are selected or until they distribute their ratings about the typical fickle agent they have done maintenance for. Hence, the curve goes down in a fluctuated manner until all the fickle agents lose their credibility and never get selected, which happens in P_3 . In a similar way, Figure 6 illustrates the good agent selection percentage versus the number of runs. This graph is the complementary of the one shown in Figure 5 as the less fickle providers are selected, the more good providers are recognized. As a result of maintenance, the CRM agents would then enhance their performance since good providers are always selected.

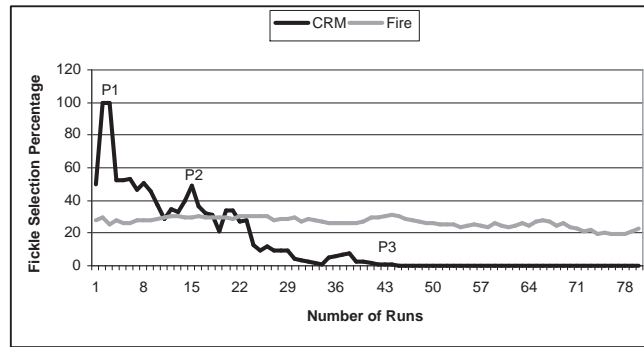


Figure 5. Comparison of CRM and FIRE in terms of selecting fickle service providers along the elapsing runs in a biased environment

In this section, we also analyze the CRM behavior compared with BRS and Travos, which are similar to CRM in the sense that they do consider other agents' suggestions while evaluating the trust of some specific agents (service providers) and discard inaccurate suggestions aiming to get adapted with the environment inconsistency attitude. In BRS, the trustor agent evaluates the recommender agents' suggestions using the beta distribution method and ignores the suggestions that deviate the most from the majority of ratings. BRS is in fact a relatively static trust method, which causes a low-efficient performance in very dynamic, open and biased environments. Cumulative gained utility vs. number of runs is shown in Figure 7. In this graph, all the agents consider the history of interactions in their selections. We should notice that it takes time for these agents to know the environment before starting gaining positive utilities and this explains why these utilities are negative and very low when the simulation starts. The BRS model is

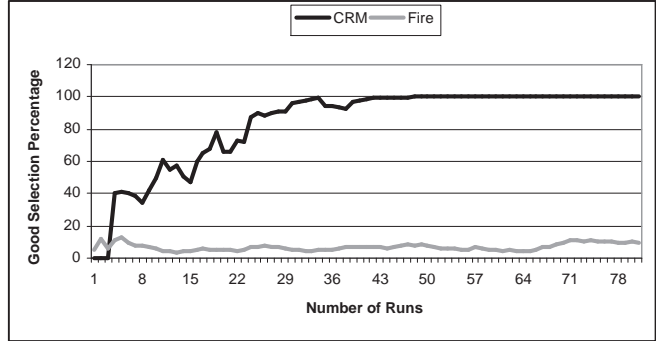


Figure 6. Comparison of CRM and FIRE in terms of selecting good service providers along the elapsing runs in a biased environment

not sensitive to an agile behavior change. This means if a BRS agent decides to evaluate a new agent, he considers the majority of ratings, which are supposed to be truthfully revealed about the trustee agent. In the case where the trustee agent has just changed his strategy, the trustor agent would lose in trust assessment and would not maintain any action to verify the accuracy of the gained information. It may take a long time that other agents perform a number of direct interactions to start rating the spurious trustee. Therefore, as illustrated in Figure 8 left plot, the BRS agents would have a higher percentage of fickle providers selection and a relatively less percentage of good providers selection (illustrated in Figure 9 left plot). The peaks in Figure 8 left plot are again a result of needing time to start knowing better the surrounding agents in the environment. It takes some while for the active agents to enhance the accuracy of their belief sets. Generally, it would take more time for the BRS agents to get adapted with the new environment conditions. The simulation results outlined in this section are all based on 50% agent activation rate.

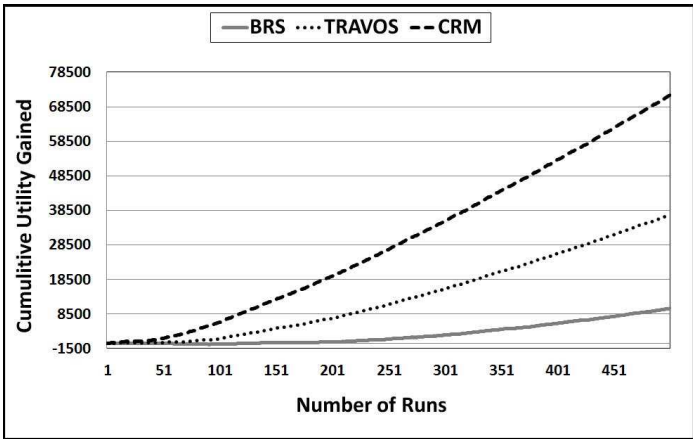


Figure 7. Comparison of CRM, Travos and BRS in terms of cumulative utility gained along the elapsing runs in a very biased environment

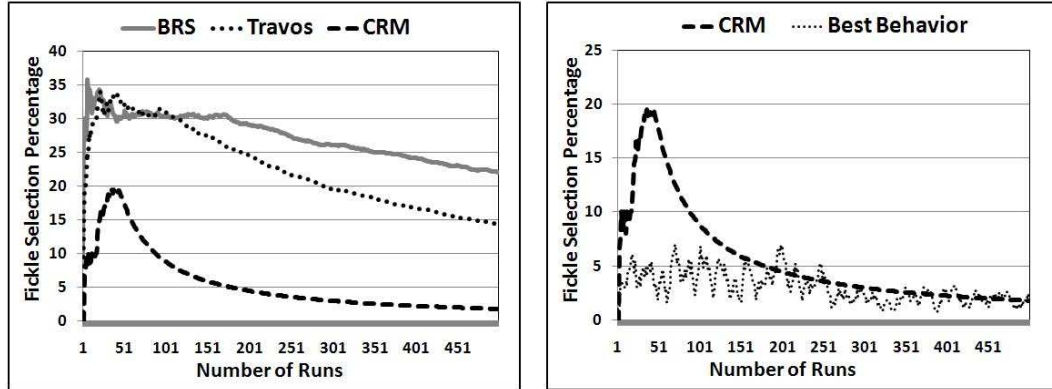


Figure 8. Left: Comparison of CRM, Travos and BRS in terms of fickle provider selection percentage along the elapsing runs with 50% activation rate in a very biased environment;
Right: Comparison of CRM with the best possible behavior in terms of fickle provider selection percentage along the elapsing runs with 50% activation rate in a very biased environment

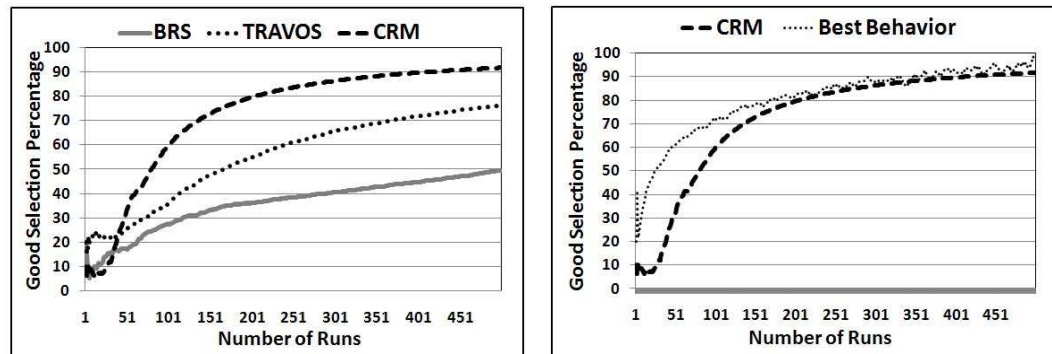


Figure 9. Left: Comparison of CRM, Travos and BRS in terms of good provider selection percentage along the elapsing runs with 50% activation rate in a very biased environment;
Right: Comparison of CRM with the best possible behavior in terms of good provider selection percentage along the elapsing runs with 50% activation rate in a very biased environment

Travos [27] has a method similar to BRS. It also uses beta distribution to estimate the trustworthiness of an agent based on the previous interaction experience. The Travos model also does not have a partial rating. It gives the trustor agent the authority to merge his own experience with recommendations from other agents. However, unlike BRS, Travos filters the surrounding agents that are fluctuating in their reports about a specific trustee agent. To some extent, this feature would implement a partial suggestion consideration and thus, the Travos agents would learn faster compared to the BRS agents. Ratings concerning the good and fickle selection percentage shown in Figures 8 and 9 left plots reflect higher efficiency of Travos compared to BRS. The Travos agents are capable of preventing the concept of fake reputation in which a group of agents artificially increase their reputation by their collusive behaviors. However, the Travos model considers that agents do not change their behaviors during runs. This unrealistic assumption affects the accuracy of trust estimation in a very biased environment. On the other

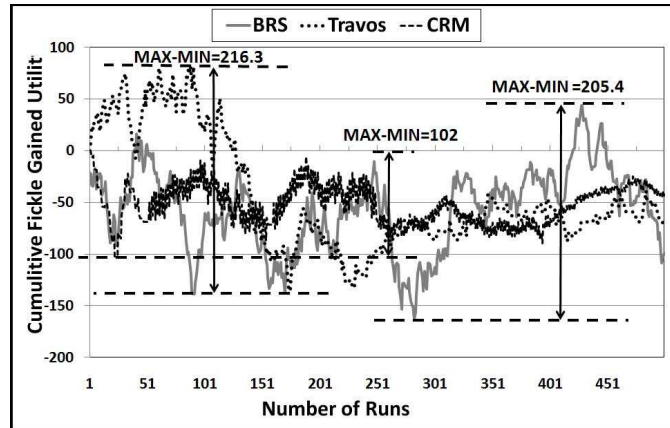


Figure 10. Comparison of CRM, Travos and BRS in terms of fickle gained utility along the elapsing runs with 50% activation rate in a very biased environment

hand, lack of agile learning ability for agents will weaken the protection against collusion and fake behaviors. 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.

The Travos and BRS trust models enable agents to sense the environment and upgrade their beliefs over time. Compared to the performance of FIRE, the Travos and BRS agents attempt to improve their best agent selection. However, these models have some aforementioned limitations that cause wrong direction to accurate trust estimation. In CRM, the aim is to improve the trust mechanism to deal with these limitations by enabling agents to get adapted while the environment is strictly intermittent. The CRM agents are equipped with the maintenance procedure by which they update their beliefs about the service providers together with the accuracy of the ratings provided by the neighbor agents in support or against a specific provider. Considering all the involved parameters, the agent that is doing maintenance balances his beliefs to be more accurate in terms of knowing the best provider and the best neighbors that can be consulted. Therefore, as shown in Figure 7, the CRM agents would gain more utility compared to the other two models. Figures 8 and 9 left plots reflect the CRM agile reaction to decrease at maximum its fickle selection percentage, and thus increase its good selection percentage very fast. In these figures, the right plots illustrate the comparison of the CRM agents with the best possible behavior in cases of fickle and good selection percentages. The best behavior is obtained by running an agent that at each run is capable of selecting the best providers that yield the maximum utility (this provider could be a fickle agent playing the role of a good agent by providing a high quality of service). As shown in these plots, the CRM selection percentages get closer to the best selection behavior, which shows the adaptability of these agents to the changes of agent behaviors. To better analyze the impact of the fickle agents that should be avoided, we have shown the gained utility from these agents in each run in Figure 10. This figure highlights the fact that the gained utility from selecting fickle agents is ideally minimized because there is no guarantee about the utility they can provide. Consequently, the high performing agents would not rely on this utility but accumulate the obtained unitively from selecting the good providers.

6. Related Work

According to [16], there are 3 main categories of trust evaluation models: *predictive models* [21,20,25], *transactional models* [12,20,27] and, *mechanism design based models* [13,36]. Predictive models have a sub-category of models using witness's information in order to increase accuracy of their trust measurement. Perhaps the best-known approaches to trust using witness's ideas in multi-agent systems are FIRE [10], SPORAS [35], Referral [33], Regret [20], the Beta Reputation System [12], and TRAVOS [27]. In this section, we analyze some recently emerged systems like TRSIM [6], Repage [21], Formal [30], and HIT [29]. We also analyze and compare some related robust frameworks like [7,26,14]. These models are worthy to be discussed in terms of their attempt to address the trust establishment in a different perspective. Predictive models also address the security establishment of the network of active agents [9,19,25]. In [9], an approach based on interpreted functions has been proposed to allow an agent to estimate the security level of message components that he receives so that he can handle them correctly. In [19], some critical issues relative to trust and reputation of multi-agent systems are introduced and analyzed. The authors propose then recommendations in order to control the environment. In [25], core design issues of trust establishment in network of dynamic agents are proposed. Moreover, different types of possible attacks together with their protection methodologies are identified and discussed.

Generally speaking, all these approaches are following a direction to implement the following guidelines: (1) the model should be provided with adequate information related to the environment and contributing agents; (2) the model should avoid consulting a central control unit that is always subject to single point of failure or huge bottleneck (for example in online auction systems). Agents are aimed to make estimation and prediction independently. The issue is that there are always malicious (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.

The idea of witness reputation has been used by Sabater who proposed a decentralized trust model called REGRET [20]. REGRET uses the reports from the witnesses in addition to the technique based on direct interaction experience. One of the substantial aspects of this work is that unlike the previous approaches, the ratings are dealt with respect to their freshness. 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 these values equally. In [7], a comparison between two trust approaches is discussed. The first approach computes the weighted trust by aggregating the past information (history) with respect to the freshness. The second approach considers some features that reflect the dynamics of trust such as the *asymmetry* property stating that trust is hard to gain but easy to lose. Their results show that the framework that takes into account the dynamics of trust outperforms the one without such a property. The dynamic approach discussed in [7] is similar to the CRM framework in the sense that the objective is to get adapted with the environment at best. However, these frameworks are different in the approach they take to address the problem of dynamism, which is considered as an

important factor in open multi-agent systems where agents can rapidly change their behaviors. In fact, CRM uses the weighed trust combined with the retrospect trust update, which allows increasing the accuracy of trust estimation by identifying the most reliable agents. However, the dynamic framework proposed in [28] (*SinAlpha*) and analyzed in [7] uses a trigonometric formula where the shape allows reflecting the properties of trust dynamics such as the *asymmetry* property.

In [30], Wang and Singh have developed an algebraic method for aggregating trust over graphs understood as webs of trust. They state that current approaches based upon combining trust reports tend to involve ad hoc formulas. So they bring up a solution from a conceptual perspective using 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 agent system where trust reports are gathered from multiple sources. In the same line of research, Jurca and Garcin in [14] propose an approach to aggregate the trust reports using incentives provided to obtain accurate reports and basically avoid the biased trust reports that are transmitted in support of an agent. In [14], the authors perform a detailed and robust analysis on the factors that influence the users that provide the feedback regarding trustworthiness of an agent. These frameworks are different from the CRM framework in the sense that the structures of trust evaluation mechanisms are different (discounting and incentives vs. quick adaptation). However, the incentive-based information collection seems to have a strong impact on the accuracy of trust and reputation establishment.

Regarding ad hoc formulation, a related work has been done by Velloso and his colleagues who assign trust levels in ad hoc networks [29]. The main characteristic of their work is that they have referred to the 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 recency relevance and relationship maturity. However, agents in this framework do not have reasoning capabilities. Moreover, they do not have policies for dealing with malicious agents. Considering the ad hoc network and human concept of trust represented in [29], there is a similar framework represented in [26] that considers the similarity (of interactions) in trust. In this framework, if an agent relies on another agent with respect to a strong history of interactions, the agent can also rely on another agent that looks alike the first one in trust features. The heuristics used in [26] are analyzed with subjective logic that enhance the agents evaluation mechanism with respect to time and number of interactions. Our framework does not consider the similarity issue, but adding it is in our future work plan.

In the work done by Shi et al. [24], 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 state that 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. However, they have not taken into account the measurements, which would unbalance the trust estimation and their decision-making procedures are solely based on the previous interactions. Unlike this approach, in our model after a certain amount of time, maintenance is performed to dynamically update the adopted policies.

In the TRSIM model [6], the authors consider two information bases (experience of trust and experience of reputation) for each agent that is evaluating trust and reputation of other agents. Different functions have been proposed to assess and update the trust and reputation values. The experience of trust is obtained from the direct interactions to which agents participate; however the reputation is the public opinion regarding a specific agent. TRSIM is equipped with some grouping criteria distinguishing reliable referees from doubtful ones. TRSIM also has an update process of trust values in which two factors of real offered quality and fulfillment of the promised quality are taken into account. The TRSIM model is close to our proposed framework in terms of aggregating trust, partitioning referee agents and performing an update process. However, the way these three components are effectively formalized and computed are completely different. Particularly, the maintenance processes in these two models address different issues. In TRSIM, the obtained experience from similar tasks helps the agent consider the right way of interacting with an agent. However, in our maintenance approach, agents are enabled to overall converge to a more accurate and knowledgeable agent interacting in dynamic multi-agent environment.

In [21], Sabater et al. have proposed Repage, a model based on ReGreT. In this model, authors have concentrated on how to select more reliable potential partners (witnesses). They have composed other agents' beliefs about the trustee agent (reputation) and agent internal belief from direct interaction (image) in order to estimate trustworthiness of the trustee agent. They have tried to preserve autonomy of agents by adding some constraints into the agents' belief sets. The proposed framework uses a fuzzy representation of reputation and its evaluation. The model's architecture is designed using the social cognitive theory. In Repage, the authors mostly concentrate on the agents' interaction, but in multi-agent systems, the environment is a particular part of the system. The environment may contain fickle agents and those have an important role to alter the other agents' decisions based on their fake behaviors. Considering the existence of fickle agents in multi-agent environments, Repage model suffers from interacting with those agents in the system. Because agents in Repage do not have a learning ability as part of their decision making process, they cannot recognize agents' fake behaviors based on their previous interactions.

7. Conclusion

The contribution of this paper is the proposition of a new probabilistic-based model to secure multi-agent systems in which agents communicate with each other using dialogue games. The trust assessment procedure is composed of on-line and off-line evaluation processes. On-line framework is based upon trustworthy and referee agents as well as several other features. Objectively, this allows enhancing the accuracy for agents to make use of the information communicated to them by other agents. Off-line framework considers the communicated information to judge the accuracy of the consulting agents in the previous on-line trust assessment process.

Our model has the advantage of being comprehensive and taking into account five important factors: (1) the trust (from the viewpoint of the trustor agents) of consulting agents; (2) the trust value assigned to trustee agents according to the point of view of consulting agents; (3) the number of interactions between consulting and trustee agents;

(4) the timely relevance of provided information; and (5) the confidence of consulting agents on the provided data. Moreover, the original process of maintenance proposed in this paper enables agents to dynamically adjust their beliefs and trustworthy community in a more efficient manner. The resulting model allows us to produce a comprehensive assessment of the agents' credibility in a software system even if the environment is very biased. The proposed mechanism accuracy is compared with other related models and discussed in details to prove the capabilities of our framework. As future work, we plan to enhance the maintenance process by considering the similarity metrics and use game theory and mechanism design approach [13] to analyze the incentives agents can have to encourage them to be more accurate.

References

- [1] E. Bagheri, R. Zafarani, and M. Barouni-Ebrahimi. Can Reputation Migrate? On the Propagation of Reputation in Multi-Context Communities. *Knowledge-Based Systems*, 22(6):410–420, 2009.
- [2] J. Bentahar, R. Alam, Z. Maamar, and N. C. Narendra. Using Argumentation to Model and Deploy Agent-based B2B Applications. *Knowledge-Based Systems*, 23(7):677–692, 2010.
- [3] J. Bentahar, Z. Maamar, D. Benslimane, and P. Thiran. An Argumentation Framework for Communities of Web Services. *IEEE Intelligent Systems*, 22(6):75–83, 2007.
- [4] J. Bentahar and J.-J. C. Meyer. A New Quantitative Trust Model for Negotiating Agents using Argumentation. *The International Journal of Computer Science and Applications*, 4(2):1–21, 2007.
- [5] J. Bentahar, F. Toni, J.-J. C. Meyer, and J. Labban. A Security Framework for Agent-based Systems. *The International Journal of Web Information Systems*, 3(4):341–362, 2007.
- [6] A. Caballero, J. Botia, and A. Gomez-Skarmeta. A New Model for Trust and Reputation Management with an Ontology Based Approach for Similarity Between Tasks. In K. Fischer et al., editor, *Multiagent System Technologies*, volume 4196 of *LNCS*, pages 172–183, 2006.
- [7] A. Danek, J. Urbano, A. Rocha, and E. Oliveira. Engaging the Dynamics of Trust in Computational Trust and Reputation Systems. In *Proc. of the 4th International KES Symposium on Agents and Multi-agent Systems Technologies and Applications, Gdynia, Poland, PP. 22-31*, 2010.
- [8] C. Dellarocas. The Digitization of Word-of-Mouth: Promise and Challenges of Online Feedback Mechanisms. *Management Science*, 49(10):1407–1424, 2003.
- [9] H. Houmani, M. Mejri, and H. Fujita. Secrecy of Cryptographic Protocols under Equational Theory. *Knowl.-Based Syst*, 22(3):160–173, 2009.
- [10] T. Huynh, N. Jennings, and N. Shadbolt. An Integrated Trust and Reputation Model for Open Multi-Agent Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
- [11] T. Huynh, N. Jennings, and N. Shadbolt. Certified Reputation: How an Agent Can Trust a Stranger. In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1217–1224, 2006.
- [12] A. Jsang and R. Ismail. The Beta Reputation System. In *Proc. of the 15th Bled Electronic Commerce Conference*, pages 324–337, 2002.
- [13] R. Jurca and B. Faltings. Obtaining Reliable Feedback for Sanctioning Reputation Mechanisms. *Journal of Artificial Intelligence Research*, 29(1):391–419, 2007.
- [14] R. Jurca, F. Garcin, A. Talwar, and B. Faltings. Reporting Incentives and Biases in Online Review Forums. *ACM Transactions on the Web (TWEB)*, 4(2):1–27, 2010.
- [15] G. Kastidou, K. Larson, and R. Cohen. Exchanging reputation information between communities: A payment-function approach. In *Proc. of the Twenty-first International Joint Conference on Artificial Intelligence*, pages 195–200, 2009.
- [16] R. Kerr. *Toward Secure Trust and Reputation Systems for Electronic Marketplaces*, 2007. M.Sc Thesis, University of Waterloo, Canada.
- [17] B. Khosravifar, J. Bentahar, M. Gomrokchi, and P. Thiran. Combined On-line and Off-line Trust Mechanism for Agent Computing. In *Proc. of the 23rd IEEE International Conference on Advanced Information Networking and Applications: AINA*, pp. 146-153, 2009.

- [18] B. Khosravifar, M. Gomrokchi, J. Bentahar, and P. Thiran. Maintenance-based Trust for Multi-Agent Systems. In *Proc. of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems: AAMAS*, pp. 1017-1024, 2009.
- [19] F. Marmol and G. Perez. Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. *Elsevier Computers and Security*, 28(7):545–556, 2009.
- [20] J. Sabater. *Trust and Reputation for Agent Societies*, 2003. Ph.D. Thesis, Universitat Autònoma de Barcelona.
- [21] 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.
- [22] M. Sensoy and P. Yolum. Ontology-Based Service Representation and Selection. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):843–857, 2004.
- [23] E. Shakshuki, L. Zhonghai, and G. Jing. An Agent-based Approach to Security Service. *International Journal of Network and Computer Applications*, 28(8):183–208, 2005.
- [24] J. Shi, G. Bochmann, and C. Adams. *A Trust Model with Statistical Foundation*, volume 173 of *IFIP International Federation for Information Processing*, pages 145–158. Springer, 2005.
- [25] Y. Sun, Z. Han, and K. Liu. Defense of trust management vulnerabilities in distributed networks. *IEEE Communications Magazine*, 46(2):112–119, 2008.
- [26] M. Tavakolifard, P. Herrmann, and S. J. Knapskog. Inferring Trust Based on Similarity with TILLIT. In *Proc. of the Third IFIP WG 11.11 International Conference on Trust Management (IFIPTM 2009)*. Springer, West Lafayette, USA., pages 133–148, 2009.
- [27] W. Teacy, J. Patel, N. Jennings, and M. Luck. Travos: Trust and Reputation in the Context of Inaccurate Information Sources. *Journal of Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.
- [28] J. Urbano, A. Rocha, and E. Oliveira. *Computing Confidence Values: Does Trust Dynamics Matter?*, volume 5816 of *Lecture Notes in Artificial Intelligence*, pages 520–531. Springer, 2009.
- [29] P. Velloso, R. Laufer, M. Duarte, and G. Pujolle. *HIT: A Human-Inspired Trust Model*, volume 211 of *IFIP International Federation for Information Processing*, pages 35–46. Springer, 2006.
- [30] Y. Wang and M. Singh. Formal Trust Model for Multiagent Systems. In *Proc. of the International Joint Conference on Artificial Intelligence*, pages 1551–1556, 2007.
- [31] L. Xiong and L. Liu. PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.
- [32] P. Yolum and M. Singh. Engineering Self-Organizing Referral Networks for Trustworthy Service Selection. *IEEE Transaction on Systems, Man, and Cybernetics*, 35(3):396–407, 2005.
- [33] B. Yu and M. Singh. An Evidential Model of Distributed Reputation Management. In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 294-301, 2002.
- [34] B. Yu and M. Singh. Searching Social Networks. In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 65-72, 2003.
- [35] G. Zacharia and P. Maes. Trust Management Through Reputation Mechanisms. *Applied Artificial Intelligence*, 14(9):881–908, 2000.
- [36] J. Zhang, R. Cohen, and K. Larson. *A Trust-based Incentive Mechanism for E-Marketplaces*, volume 5396 of *Lecture Notes in Artificial Intelligence*, pages 135–161. Springer, 2008.