International Journal of Innovative Computing, Information and Control
Volume **9**, Number **6**, June **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 2405–2420

A COMBINATION TRUST MODEL FOR MULTI-AGENT SYSTEMS

Manh Hung Nguyen[1,2] and Dinh Que Tran[1]

[1]Post and Telecommunication Institute of Technology
Ha Noi, Vietnam
{ nmhufng; tdque }@yahoo.com

[2]IRD, UMI 209 UMMISCO
Institut de la Francophonie pour l'Informatique (IFI)
Hanoi, Vietnam

Received March 2012; revised July 2012

Abstract. *Trust plays a crucial role in supporting agents to select the partners while interacting with each other in open distributed multi-agent systems. Researches on computational trust mostly focus on considering three types of trust: individual experience, inference trust from other agents in the community, and hybrid trust. Most current hybrid trust models are the combination of experience and inference trusts and make use of some propagation mechanism to enable agents to share his/her final trust with partners. However, the disadvantage of these models is that their use of such mechanism could result in high cost in computation. The problem with such a propagation based sharing may be reduced to the issue of considering which type of trusts an agent should utilize in evaluating partners and which type of trusts should be shared among agents. In this paper, we first propose a computational trust model which is a combination of experience and reference trusts. Our model may enable agents to evaluate the degree of trust on their partners, which is computed with a weighted combination based on the linguistic quantifier function of their own experiences and the trustworthiness on their partners in the agent society. Then, we investigate which types of trust should be shared among agents and how they affect the effectiveness of interaction in our model. Our experiments are performed in an e-commerce environment and the experimental results have demonstrated that: (i) Changing two internal parameters, which are the linguistic quantifier functions and weight vectors for aggregating the experience and reference trusts, may contribute to selecting the best partner for an agent but not much; (ii) combining the experience and inference trusts is better than using the single experience trust for estimating the trustworthiness of a partner; (iii) agents may share their single experience trust or combination trust with their partners and furthermore their making use of any of these sharing strategies may not affect the evaluation of product quality.*

Keywords: Multi-agent system, Autonomous agent, Interaction, Trust, Reputation

1. Introduction.

Many software applications are open distributed systems whose components are decentralized, constantly changed, and spread throughout network. Peer-to-peer networks, web service, semantic web, social network, recommender systems in e-business, autonomic and pervasive computing are among such systems. These systems can be modeled as multi-systems that are composed of autonomous agents interacting with each other according to some communication mechanisms and protocols. The problem of how agents decide with whom and when to interact has become the active research topic in the recent years (e.g., [2, 4, 5, 6, 8, 14]). It means that they need to deal with degrees of uncertainty in making decisions during their interaction. Trust among agents is the most important foundation based on which agents decide to interact with other ones. And the problem of making decision in interaction results in the one of estimation of their own trust on their partners. The more trust a partner is evaluated, the more possibility with the partner agents decide to interact.

Trust has been approached by researchers with various points of view (Gambetta [2], Josang et al. [5]). From the computational viewpoint, trust is defined as a quantified belief by a truster with respect to the competence, honesty, security and dependability on a trustee within a specified context (Grandison and Sloman [3]). Such definition of the computational trust will be basis for the model of trust presented in this paper. There are three main types of computational trust being considered in multiagent systems: (i) Experience based trust; (ii) reference based trust (Reputation); (iii) hybrid computation based trust.

Firtly, models of trust based on *personal experience* compute the trust value of truster on trustee based on truster's personal parameters during transactions that have been performed with the trustee in the past. For instance, Manchala [7] and Nefti et al. [8] proposed models for the trust measure in e-commerce based on fuzzy computation with parameters such as cost of a transaction, transaction history, customer loyalty, indemnity and spending patterns. The probability theory-based model of Schillo et al. [11] is intended for scenarios where the result of an interaction between two agents is a boolean impression such as good or bad but without degrees of satisfaction. Shibata et al. [13] used a mechanism for determining the confidence level based on agent's experience with Sugarscape model, which is artificially intelligent agent-based social simulation.

Secondly, models of trust based on *reference* (also called reputation) estimate trust value based on truster's reference with her/his partners in their community or system. The basic principle of this approach is that the more an agent is trusted by many other agents, the more trustworthy s/he is. For instance, in the model proposed by Yu and Singh [16], only the information stored by an agent about the most recently direct interactions is involved in computation. The most recent experiences with each concrete partner are considered for the calculations. Li and Gui [6] proposed a reputation model based on human cognitive psychology and the concept of direct trust tree (DTT).

Thirdly, *hybrid models* utilize an integration of both personal experience and reference trust by means of weighted averaging operators. For instance, the trust models proposed by Sen and Sajja [12] make use of two trust acquisition mechanisms, which based on observation and interaction. The main idea behind the reputation model presented by Carter et al. [1] is that "the reputation of an agent is based on the degree of fulfillment of roles ascribed to it by the society". The hybrid model ReGreT introduced by Sabater and Sierra [10] is intended for complex small/mid-size e-commerce environments, where social relations among individuals play an important role. Ramchurn et al. [9] developed a trust model based on fuzzy sets for computing confidence and reputation. They also show how it can be applied to guiding agents in evaluating past interactions and in establishing new contracts with one another. The hybrid model FIRE proposed by Huynh et al. [4] integrates a number of information sources to produce a comprehensive assessment of an agent's likely performance in open systems. And Victor et al. [14] advocate the use of a hybrid trust model with score couples (trust, distrust), which may be drawn from gradual trust, distrust, ignorance, and inconsistency of information.

Most current hybrid models take advantages of experience trust (confidence) and reference trust (reputation). However, two problems need to be investigated furthermore. (i) Whether agents need to utilize combination of two trust metrics or only one of two trust types for selection of partners in cooperation. (ii) The current hybrid models make use of propagation mechanisms for computing reference trust. However, the application of these mechanisms may result in high cost in computation. The problem may be reduced to the question whether agents should share combination trust or only need to dispatch experience trust.

In this paper, we propose a computational model of trust which is a combination of experience and reference trusts. Our hybrid model makes use of fuzzy computational techniques and weighted aggregation operators for integration. The contributions of our work are three-fold:

- Identifying problems with current trust models and proposing a new hybrid model that addresses all of these problems;
- Formalizing the model and presenting an algorithm for updating trust;
- Performing experiments to demonstrate:
  – How parameters affect trust computation results in our proposed model;
  – Agents should utilize combination trust in evaluating partners whereas they need share only experience trust with their partners during their interaction.

The remainder of this paper is organized as follows. Section 2 and Section 3 describe respectively the computational model of trust and the algorithm of trust updating. Section 4 presents the experimental evaluations of the effect of the model internal parameters on final results and of the advantages of our model regarding other related works. Section 5 is some discussion about our model features. Final section is our conclusions and future works.

2. **Computational Model of Trust.**

2.1. **Model of trust.** Consider a typical system as peer-to-peer (P2P) in which all peer agents are legal and suppose that necessary protocols for their exchanging the information involving trust are always available. Our computational trust model in a multi-agent system is represented by a 4-tuples (A, E, R, T), in which:

- $A = \{1, 2, \dots, n\}$ – a set of agents taking part in the system;
- $E = \mid A \mid * \mid A \mid$ – a matrix of experience trust. Elements E_{ij} represent the experience trust an agent i has on an agent j.
- $R = \mid A \mid * \mid A \mid$ – a matrix of reference trust. Elements R_{ij} represent the reference trust an agent i has on an agent j.
- $T = \mid A \mid * \mid A \mid$ – a matrix of overall trust. Elements T_{ij} represent the overall trust an agent i has on an agent j. The greater the value T_{ij} is, the more an agent i trusts on an agent j.

Following sections will formalize the computational model for estimating the values of E_{ij}, R_{ij} and T_{ij}.

2.1.1. *Experience trust.* Intuitively, experience trust of agent i on agent j is the trustworthiness about j that agent i collects from all transactions between i and j in the past. Let U_{ij} be a set of transactions having been performed between agent i and agent j until the current time. Transactions are timely ordered. For instance, transaction k is the k^{th} latest transaction between i and j.

Definition 2.1. *Transaction trust t_{ij}^k, where $t_{ij}^k \in [0, 1]$, $k = 1, \dots, \mid U_{ij} \mid$ and $i, j = 1, \dots, n$ is the trustworthiness of agent i about agent j from the k^{th} latest transaction between i and j. The value of t_{ij}^k is determined as 1 if positive trust, 0.5 if neutral, 0 if negative trust.*

Definition 2.2. *Vector t_{ij} representing trustworthiness of all transactions, ordered in time, between agent i and its partner j, is called a transaction trust vector. Denote $t_{ij} = (t_{ij}^k)$, where $k = 1, \dots, \mid U_{ij} \mid$ and $i, j = 1, \dots, n$.*

Definition 2.3. *Vector $w = (w_1, w_2, \ldots, w_{|U_{ij}|})^T$ is called the transaction weight vector if $w_k \in [0, 1]$, $k = 1, \ldots, | U_{ij} |$, is the weight of the k^{th} latest transaction based on evaluation given agent i such that:*

$$\begin{cases} w_{k_1} \geqslant w_{k_2} \text{ if } k_1 < k_2 \\ \sum_{k=1}^{|U_{ij}|} w_k = 1 \end{cases} \tag{1}$$

The weight vector is decreasing from head to tail because the aggregation focuses more on the later transactions and less on the older transactions. It means that the later the transaction is, the more its trust is important for estimating the experience trust of the correspondent partner. This vector may be computed by means of Regular Decreasing Monotone (RDM) linguistic quantifier (Zadeh [17], Yager [15]) as follows:

Definition 2.4. *The function $Q : [0, 1] \to [0, 1]$ is a Regular Decreasing Monotone linguistic quantifier, denote RDM, iff it satisfies the following conditions:*

$$\begin{aligned} (i)&\ Q(0) = 1 \\ (ii)&\ Q(1) = 0 \\ (iii)&\ Q(i_1) \geqslant Q(i_2) \text{ if } i_1 < i_2. \end{aligned}$$

It is easy to prove the following propositions:

Proposition 2.1. *The following functions are RDM:*

$$\begin{aligned} (a)&\ Q(x) = (1 - x)^m \text{ with } m \geq 1 \\ (b)&\ Q(x) = 1 - \sqrt{1 - (1 - x)^2}. \end{aligned}$$

Proposition 2.2. *Suppose that Q is a RDM function. The vector w_i generated by Q*

$$w_i = Q\left(\frac{i-1}{| U_{ij} |}\right) - Q\left(\frac{i}{| U_{ij} |}\right) \text{ for } i = 1, \ldots, | U_{ij} | \tag{2}$$

is the transaction weight vector.

Definition 2.5. *Experience trust of agent i on agent j is a mapping $f_e : [0, 1]^{|U_{ij}|} \times [0, 1]^{|U_{ij}|} \to [0, 1]$, which is defined by the formula*

$$E_{ij} = f_e(t_{ij}, w) = t_{ij} * w = \sum_{k=1}^{|U_{ij}|} t_{ij}^k * w_k \tag{3}$$

where t_{ij}^k is the trust of transaction k^{th} of agent i on its partner j and w_k is the weight of the transaction.

This is an instance of the Ordered Weighted Averaging (OWA) operators (Yager [15]) over the set of transaction trust between agent i and agent j. This operator has been widely used in several application of decision support and control.

Definition 2.6. *Vector $E_i = (E_{ij})$, $j = 1, \ldots, n$, representing the experience trust of agent i on all their partners, is called the experience trust vector.*

2.1.2. *Reference trust.* Reference trust (also called reputation trust) of agent i on agent j is the trustworthiness on agent j given by other agents in the system. We suppose that any agent can refer all agents (friend agents) she/he knows in the system about their experience trust on the other agents.

Suppose that j is an agent which the agent i has not yet interacted with but needs to evaluate to cooperate with. Let $V_{ij} \subseteq \mathscr{A}$ be a set of agents that an agent i knows and have had transactions with j in the past. Denote r_{ij}^{l} the individual reference trust of agent i on agent j via agents l $(l \in V_{ij})$.

Definition 2.7. *Individual reference trust $r_{ij}^{l} \in [0,1]$, $l \in V_{ij}$, represents the reference trust of agent i on agent j via agent l:*

$$r_{ij}^{k} = E_{kj} \tag{4}$$

where E_{lj} is the current experience trust of l on j which is determined from Definition 2.5.

Definition 2.8. *Reference trust $f_r : [0,1]^{|V_{ij}|} \rightarrow [0,1]$ is a mapping (as a non-weighted average) from all experience trust of agents, which have had transaction with agent j and i knows, into the reference trust of agent i on agent j:*

$$R_{ij} = f_r\left(r_{ij}^{1}, r_{ij}^{2}, \ldots, r_{ij}^{|V_{ij}|}\right) = \begin{cases} \dfrac{\sum\limits_{l \in V_{ij}} r_{ij}^{l}}{\mid V_{ij} \mid} & \textit{if } V_{ij} \neq \varnothing \\ 0 & \textit{otherwise} \end{cases} \tag{5}$$

Definition 2.9. *Reference trust vector $R_i = (R_{ij})$, represents the reference trust of agent i on all other agents in the system.*

2.1.3. *Overall trust.* Resulting from these partial trust measures, we may construct a definition of combination of two trusts. Intuitively, the combination trust of experience and reference trusts must satisfy the following conditions:

- It must neither lower than the minimal and nor higher the maximal of experience trust and reference trust;
- The more the experience trust is high, the more the combination trust is high;
- The more the reference trust is high, the more the combination trust is high.

These constraints may be represented by the following *combination function*:

Definition 2.10. *A function $t : [0,1] \times [0,1] \rightarrow [0,1]$ is called the combination function, denote com-function, if and only if it satisfies the following conditions:*

1. $\min(e,r) \leqslant t(e,r) \leqslant \max(e,r)$;
2. $t(e_1,r) \leqslant t(e_2,r)$ *if* $e_1 \leqslant e_2$;
3. $t(e,r_1) \leqslant t(e,r_2)$ *if* $r_1 \leqslant r_2$.

Proposition 2.3. *Given a weight vector $w = (w_{ie}, w_{ir})$, $w_{ie} + w_{ir} = 1$. A function $t : [0,1] \times [0,1] \rightarrow [0,1]$ defined by the formula*

$$t(x,y) = w_{ie} * x + w_{ir} * y$$

is a com-function.

Definition 2.11. *Combination trust T_{ij} of agent i on agent j is defined by the formula:*

$$T_{ij} = t(E_{ij}, R_{ij}) \tag{6}$$

in which t is a com-function and E_{ij}, R_{ij} are experience and reference trust, respectively.

3. **Updating Trust.** Agent i's trust on agent j can be changed in all i life-time whenever there is at least one of these conditions occurs (as showed in Algorithm 1, line 2):

- There is a new transaction between i and j occurring (line 3), so the experience trust of i on j will be changed.
- There is an agent l who sends to i a his new experience trust about j for reference (line 10). Thus, the reference trust of i on j will be changed.

Algorithm 1 Trust updating algorithm

1: **for all** agent i in the system **do**
2:   **if** (there is a new transaction k-th with agent j) or (there is a new reference trust E_{lj} from agent l about agent j) **then**
3:     **if** there is a new transaction k with agent j **then**
4:       $t_{ij}^k \leftarrow 0$ or 0.5 or 1 // 0: negative; 0.5: neutral; 1: positive
5:       $t_{ij} \leftarrow t_{ij} \cup t_{ij}^k$ // add t_{ij}^k into t_{ij}
6:       $t_{ij} \leftarrow Sort(t_{ij})$ // re-sort t_{ij} on descending of time
7:       $w \leftarrow GenerateW(k)$ // generate the weight vector w of size k
8:       $E_{ij} \leftarrow \sum_{h=1}^{k} t_{ij}^h * w_h$ // update the experience trust
9:     **end if**
10:     **if** there is a new reference trust E_{lj} from agent l about agent j **then**
11:       **if** l is a new friend **then**
12:         $V_{ij} \leftarrow V_{ij} \cup \{l\}$ // add l into friends set V_{ij}
13:       **end if**
14:       $r_{ij}^l \leftarrow E_{lj}$ // update the individual reference trust
15:       $R_{ij} \leftarrow \dfrac{\sum_{h \in V_{ij}} r_{ij}^h}{\mid V_{ij} \mid}$ // update the reference trust
16:     **end if**
17:     $T_{ij} \leftarrow t(E_{ij}, R_{ij})$ // update the final trust
18:   **end if**
19: **end for**

E_{ij} is updated after the occur of each new transaction between i and j as follows (lines 3 – 9):

- The new transaction's trust value t_{ij}^k is placed at the first position of vector t_{ij} (lines 4 – 6). Function $Sort(t_{ij})$ sorts the vector t_{ij} in ordered in time.
- Vector $\vec{w}$ is also generated again by applying Formula (2) with the size of $k = \mid U_{ij} \mid$ (line 7) in function $GenerateW(k)$.
- E_{ij} is updated by applying Formula (3) with the new vector t_{ij} and w (line 8).

Once E_{ij} is updated, agent i sends E_{ij} to all friend gents. Therefore, all i's friends will update their reference trust when they receive E_{ij} from i. We suppose that all friend relations in system are bilateral, this means that if agent i is a friend of agent j then j is also a friend of i.

After having received E_{lj} from agent l, agent i then updates her/his reference trust R_{ij} on j as follows (lines 10 – 16):

- In the case l is the new agent whom agent i knows, agent l will be also added into V_{ij} (lines 11 – 13).

- In order to update the individual reference trust r^l_{ij}, the value of E_{lj} is placed at the position of the old one (line 14).
- R_{ij} is updated by applying Formula (5) with the new set V_{ij} (line 15).

Finally, T_{ij} is updated by applying Formula (6) from new E_{ij} and R_{ij} (line 17).

4. A Case Study: An E-Commerce System.

4.1. A scenario of the system.

In this section, we consider a scenario of an e-commerce system to study how the parameters in the model effect on the results, and then compare the proposed model with others.

An e-market is composed of a set of seller agents, a set of buyer agents, and a set of transactions. Each transaction is performed by a buyer agent and a seller agent. A seller agent plays the role of a seller who owns a set of items and it could sell many items to many buyer agents. A buyer agent plays the role of a buyer who could buy many items from many seller agents.

Each buyer agent has a transaction profile to calculate the experience trust of all seller agents having transaction(s) with it. It has also a set of reference trusts (reputation) about seller agents which are referred from other buyer agents. When a transaction is made between a buyer agent and a seller agent, the buyer agent will evaluate the transaction based on the quality of the received item. The buyer agent will then update its trust on its seller agents once it finishes a transaction or receives a reference trust from other buyer agents. The buyer agent selects the most trustworthy seller agent by applying our proposed model when it intends to buy a new item.

4.2. Effects of internal parameters.

This model has two internal parameters. First, the linguistic quantifier Q which generates the weight vector w in order to aggregate all transaction trusts into experience trust. Second, the weight pair $< w_{ie}, w_{ir} >$ which aggregates the experience trust and reference trust into general (final) trust. In this section, we examine the effect of each such a parameter on our model.

4.2.1. *Effects of linguistic quantifier Q.* The scenario is composed of four steps as follows:

- *Step 1:* Using the same input data for all simulations. The input data includes the set of transactions, the buyer agent, the seller agent and the item for each transaction.
- *Step 2:* Applying the proposed model with several linguistic quantifier Q. For comparison, we make use of four functions: $(1-x)$, $(1-x)^2$, $(1-x)^4$ and $(1-x)^8$.
- *Step 3:* Saving the output for each time of simulation. The output is a vector of size n (n is number of buyer agents) representing the best partner of each buyer agent: $O[i] = j$ means that the best seller agent of buyer agent i is j (the one whom i has the highest final trust on).
- *Step 4:* Comparing the similarity among obtained outputs.

The main criterion to compare, in this experiment, is the degree of used function Q derivative. It features the distribution of weights in weight vector w. The more degree of used Q function derivative is high, the more the vector weight w centralizes at the beginning of it. Therefore, the latest transaction is more important. Figure 1 shows results of comparison. The vertical axe represents the average percentage of outputs similarity. The horizontal axe represents the difference between the degree of used function Q derivatives. For instance, such a difference between $(1-x)$ and $(1-x)^4$ is 3, between $(1-x)^2$ and $(1-x)^8$ is 6.

The chart shows that there is a linear-like effect of linguistic quantifier Q on the output. Thus the more the difference between the degree of used function Q derivatives, the more outputs are different. However, this difference is not so much. Even in the case of its

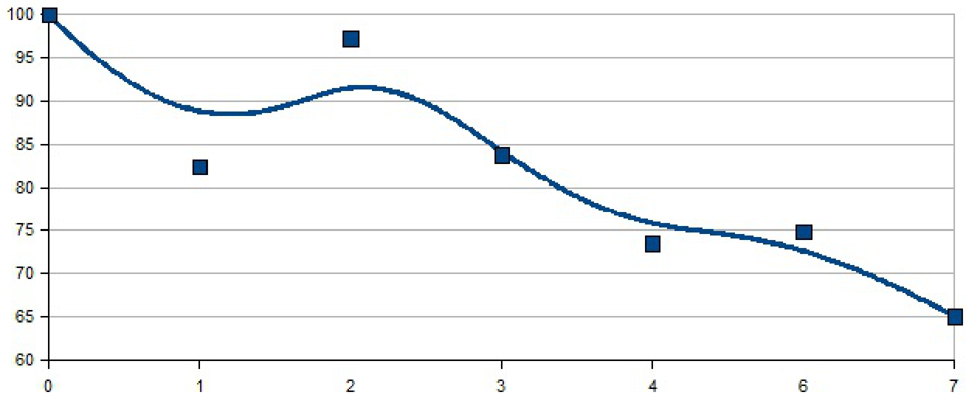


FIGURE 1. Similarity on output featured by the difference between linguistic quantifier Qs

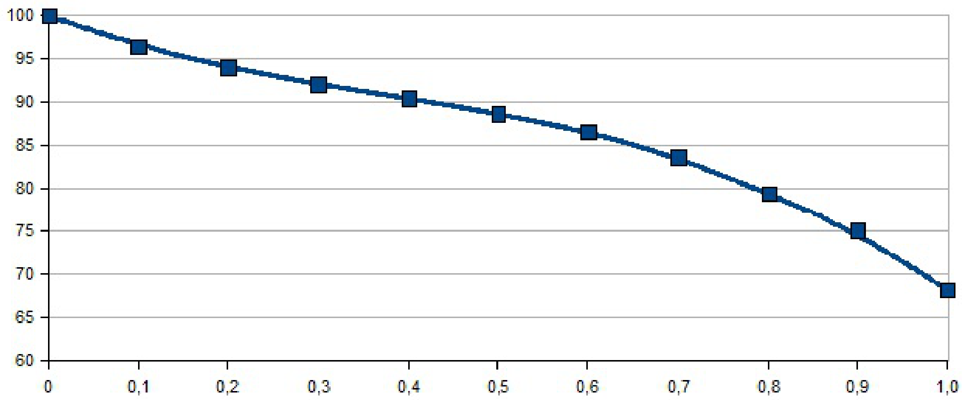


FIGURE 2. Similarity on output featured by the difference between weight pair $< w_{ie}, w_{ir} >$

maximum, the output similarity remains more than 60%. In conclusion, we can say that the change on linguistic quantifier Q can contribute to the selection of the best partner of certain agent in the system, but the contribution is not much.

4.2.2. *Effects of weight pairs $< w_{ie}, w_{ir} >$.* We use the same method as in the test of linguistic quantifier Q, except that in Step 2, we run the algorithm for eleven selected weight pairs as: $< 1.0, 0.0 >$, $< 0.9, 0.1 >$, $< 0.8, 0.2 >$, $< 0.7, 0.3 >$, $< 0.6, 0.4 >$, $< 0.5, 0.5 >$, $< 0.4, 0.6 >$, $< 0.3, 0.7 >$, $< 0.2, 0.8 >$, $< 0.1, 0.9 >$ and $< 0.0, 1.0 >$.

The main criterion to compare in this experiment is the difference between weight pairs. Figure 2 shows results of comparison. The vertical axe represents the average percentage of output similarity. The horizontal axe represents the difference between weight pairs. For instance, such a difference between $< 0.9, 0.1 >$ and $< 0.3, 0.7 >$ is 0.6, between $< 0.6, 0.4 >$ and $< 0.5, 0.5 >$ is 0.1.

The chart shows that there is a linear-like effect of weight pair $< w_{ie}, w_{ir} >$ on the output. The more the difference between the used weight pair $< w_{ie}, w_{ir} >$, the more outputs are different. However, like the case of linguistic quantifier Q, this difference is not too much. Even in the case of its maximal, the output's similarity remains more than 60%. Therefore, we can say that the change on weight pair $< w_{ie}, w_{ir} >$ can contribute to the selection of the best partner of certain agent in the system, but it is not much.

4.3. **Comparison with other models.**

4.3.1. *Experimental setup.* In our experiment, two questions need to be answered:

- First, it is better if a buyer agent performs its processing based on the combination of both experience and reference trust or based only on one of trust types.
- Second, it is better if buyer agent shares its final trust or shares only its experience trust on its seller.

Initial Parameters

In order to make the results comparable and to avoid the effect of randomness in initial values of simulation parameters, we use the same values for input parameters in all simulation scenarios: number of sellers, number of products, number of simulations. These values are given in Table 1.

TABLE 1. Simulation parameters

Parameters	**Values**
Number of runs for each scenario	100 (times)
Number of sellers	1000
Number of buyer	200
Number of products	50000
Average number of bought products/buyer	5 | 15 | 30 | 50 | 70 | 100
Average number of friends/buyer	50 | 100 | 150 | 200

Analysis and Evaluation Criteria

For each scenario, we will run 100 times. In the output, we need to calculate the following parameters:

- The average quality (in %) of brought products for all buyers. A scenario (strategy) is considered better if it brings the higher average quality of brought products for all buyers in the system.

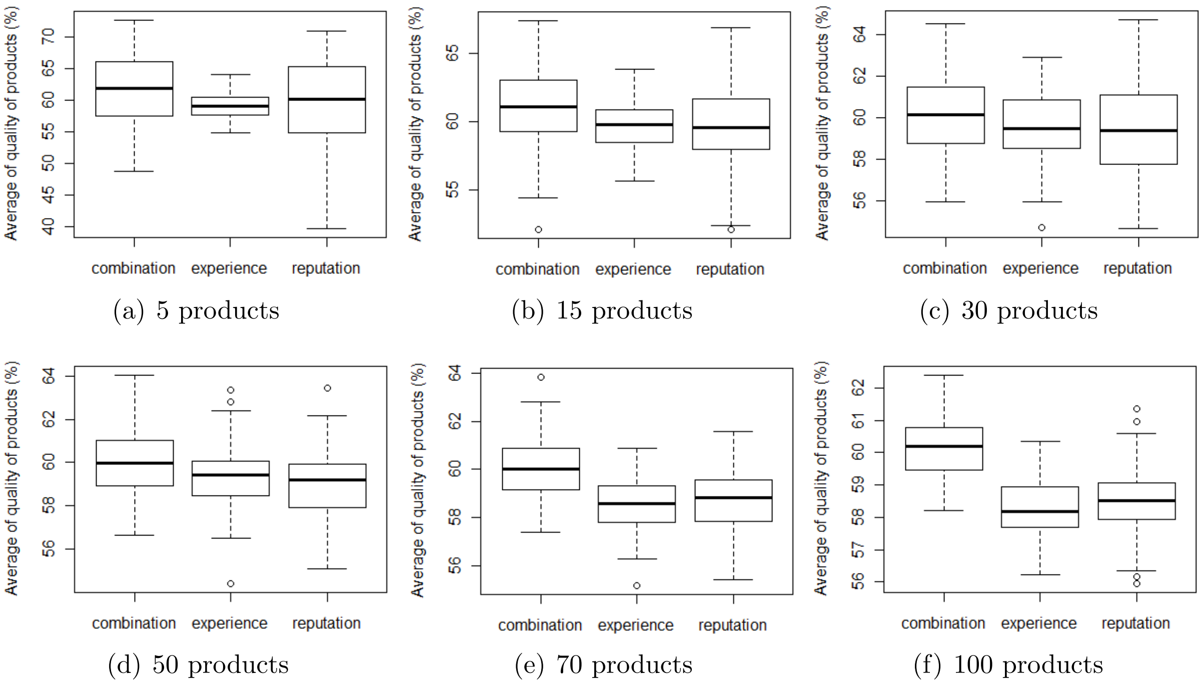
4.3.2. *Effect of combination trust and single experience or reference trust: a comparison.* The question is that in evaluating partners a buyer agent should base on the combination of both experience and reference trusts or base only either on experience trust or reference trust. In order to answer this question, we will perform a simulation with three strategies:

- *Strategy 1 – Based on combination trust*: Buyer agent chooses the most trustworthy seller based on its final trust (the combination of experience and reference trust) of its sellers.
- *Strategy 2 – Based on experience trust*: Buyer agent chooses the most trustworthy seller based only on its experience trust of its sellers.
- *Strategy 3 – Based on reference trust*: Buyer agent chooses the most trustworthy seller based only on its reference trust (reputation) of its sellers.

Intuitively, the quality of estimation of experience and reference trust could be affected by the number of items that each buyer has bought. In order to avoid the effects of this factor, we launch the simulations for each scenario in multi-cases in which each buyer agent purchases about 5, 15, 30, 50, 70 and 100 items (we use a constant number of friends – about 100 (50% of all buyers), for each buyer).

Results

When the average number of products each buyer buys is 5 (Figure 3(a)). The average quality of bought products for all buyers in case of basing on combination trust is significantly higher than in case of basing either only on experience trust ($M(combination) = 60.93\%$, $M(experience) = 59.24\%$, significant difference with *p-value* < 0.001), or only on reference trust ($M(combination) = 60.93\%$, $M(reference) = 59.71\%$, significant difference with *p-value* < 0.008).



(a) 5 products (b) 15 products (c) 30 products

(d) 50 products (e) 70 products (f) 100 products

FIGURE 3. Significant difference of average quality of products of all buyers among three strategies: based on combination trust, based on experience trust, and based on reference trust

When the average number of products each buyer buys is 15 (Figure 3(b)). The average quality of bought products for all buyers in the strategy based on combination trust is significantly higher than in the strategy based either only on experience trust ($M(combination) = 60.88\%$, $M(experience) = 59.68\%$, significant difference with *p-value* < 0.001), or only on reference trust ($M(combination) = 60.88\%$, $M(reference) = 59.53\%$, significant difference with *p-value* < 0.002).

When the average number of products each buyer buys is 30 (Figure 3(c)). The average quality of bought products for all buyers in the strategy based on combination trust is significantly higher than in the strategy based either only on experience trust ($M(combination) = 60.16\%$, $M(experience) = 59.64\%$, significant difference with *p-value* < 0.04), or only on reference trust ($M(combination) = 60.16\%$, $M(reference) = 59.39\%$, significant difference with *p-value* < 0.009).

When the average number of products each buyer buys is 50 (Figure 3(d)). The average quality of bought products for all buyers in the strategy based on combination trust is significantly higher than in the strategy based either only on experience trust ($M(combination) = 60.02\%$, $M(experience) = 59.37\%$, significant difference with *p-value* < 0.003), or only on reference trust ($M(combination) = 60.02\%$, $M(reference) = 58.93\%$, significant difference with *p-value* < 0.001).

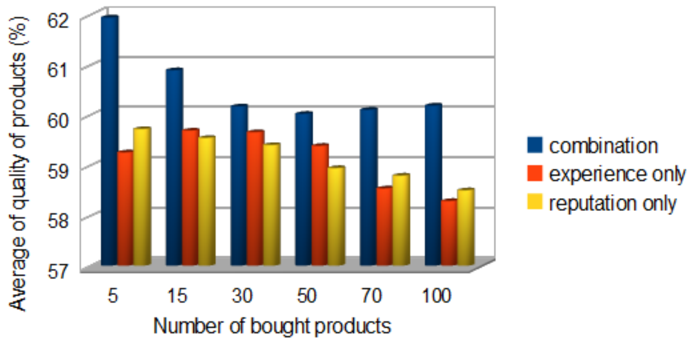


FIGURE 4. Summary of evaluating parameters among three strategies: based on combination trust, based on experience trust, and based on reference trust

When the average number of products each buyer buys is 70 (Figure 3(e)). The average quality of bought products for all buyers in the strategy based on combination trust is significantly higher than in the strategy based either only on experience trust ($M(combination) = 60.10\%$, $M(experience) = 58.53\%$, significant difference with $p\text{-}value < 0.001$), or only on reference trust ($M(combination) = 60.10\%$, $M(reference) = 58.78\%$, significant difference with $p\text{-}value < 0.001$).

When the average number of products each buyer buys is 100 (Figure 3(f)). The average quality of bought products for all buyers in the strategy based on combination trust is significantly higher than in the strategy based either only on experience trust ($M(combination) = 60.18\%$, $M(experience) = 58.28\%$, significant difference with $p\text{-}value < 0.001$), or only on reference trust ($M(combination) = 60.18\%$, $M(reference) = 58.49\%$, significant difference with $p\text{-}value < 0.001$).

Figure 4 summaries the comparison of the average quality of products of all buyers among three strategies: based on combination trust, based either on experience, based on reference trust. The results suggest that it is better if buyer bases on the final trust to estimate the seller trust than bases only on either experience trust or reference trust.

4.3.3. *Sharing the final trust or only experience trust: no difference.* There are two strategies for buyer agents sharing their trust with partners: sharing their final trust and sharing only its experience trust. The question is that which strategy is better for sharing trust among agents. In order to answer this question, we perform simulation for two strategies:

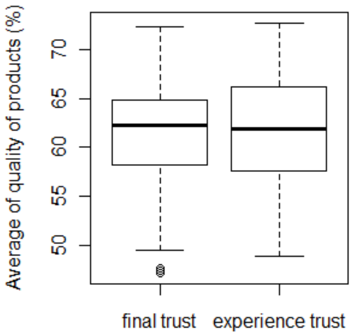
- *Strategy 1 – sharing final trust*: Buyer agent shares its final trust (the combination of experience and reference trust) of its sellers to its friends (other buyer agents).
- *Strategy 2 – sharing experience trust*: Buyer agent shares only its experience trust of its sellers to its friends.

In order to avoid the affect of the number of bought products or the number of friends of each buyer, we launch the simulations in various values of number of bought products (5, 15, 30, 50, 70, 100 – with the same number of friends is 100) and number of friend for each buyer (50, 100, 150, 200 – with the same number of bought products if 50)[1].

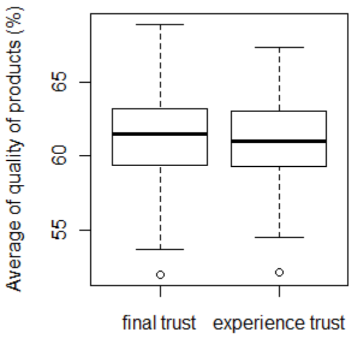
Results

The results indicate that there is no significant difference of the average quality of product from all buyers on various number of products (Figure 5) in two cases: sharing the final trust and sharing only the experience trust. In the case of 5, 15, 30, 50, 70, and 100

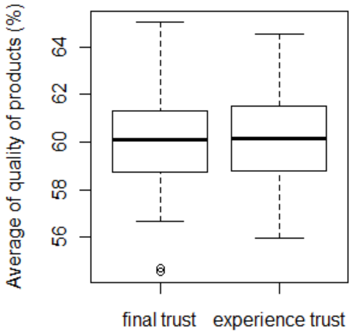
[1]We do not simulate the case in which buyer agent has no friend, because it is exactly the case that buyer agent bases only on the experience trust.



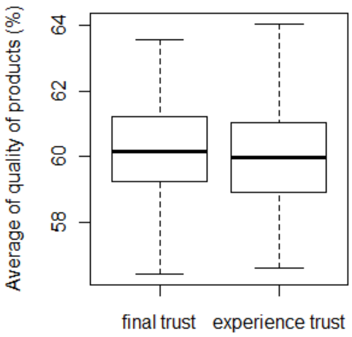
(a) 5 products



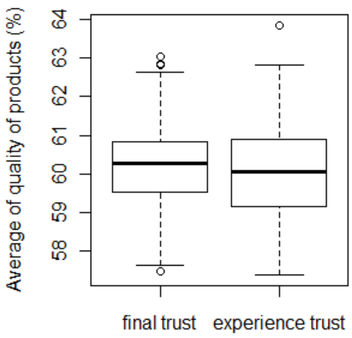
(b) 15 products



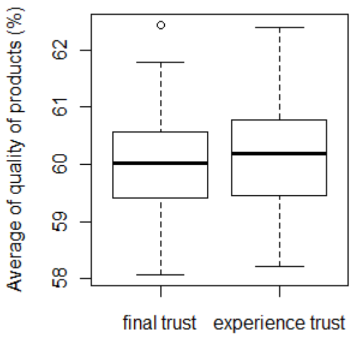
(c) 30 products



(d) 50 products



(e) 70 products



(f) 100 products

FIGURE 5. Non-significant difference of average quality of products of all buyers on various number of bought products between two strategies: sharing the final trust and sharing only the experience trust

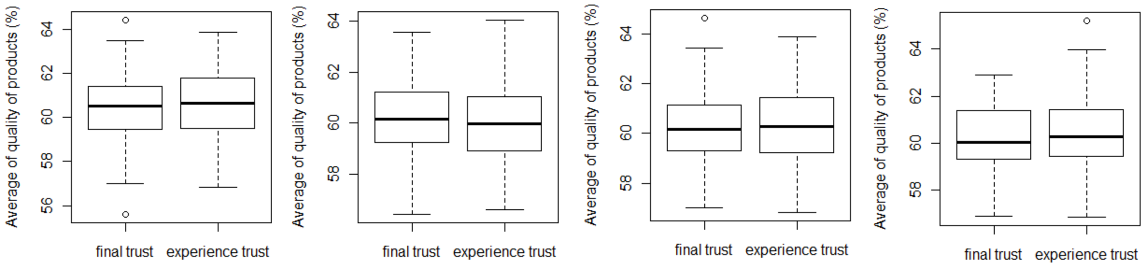
TABLE 2. Comparison of average quality of products of all buyers, on various number of bought products

N. of products	M (final trust)	M (experience trust)	*p-value*
5	61.57	61.93	>0.63
15	61.28	61.88	>0.39
30	60.08	60.16	>0.74
50	60.23	60.02	>0.30
70	60.25	60.10	>0.36
100	60.01	60.18	>0.18

bought products, the significant difference are respectively $p\text{-}value > 0.63$, $p\text{-}value > 0.39$, $p\text{-}value > 0.74$, $p\text{-}value > 0.30$, $p\text{-}value > 0.36$, and $p\text{-}value > 0.18$[2] (Table 2).

Surprisingly, the results also indicate that there is no significant difference between the average quality of products of all buyers on various number of friends of buyers in two strategies: the case of sharing the final trust and the case of sharing only the experience trust (Figure 6). In the case of 50, 100, 150, and 200 friends for buyer, the significant difference are respectively $p\text{-}value > 0.67$, $p\text{-}value > 0.31$, $p\text{-}value > 0.60$, and $p\text{-}value >$ (Table 3).

[2]Note that in the statistical technique, when we use the *t-test* for two sets A and B (*t.test(A,B)*) to consider the significant difference *p-value*: if $p\text{-}value < 0.05$ we can conclude that two sets A and B are significantly different; otherwise, we can conclude that two sets A and B are not significantly different.



(a) 50 friends (b) 100 friends (c) 150 friends (d) 200 friends

Figure 6. Non-significant difference of average quality of products of all buyers between two strategies: sharing the final trust and sharing only the experience trust, on various number of friends of buyers

Table 3. Comparison of average quality of products of all buyers, on various number of buyer's friends

N. of friends	M (final trust)	M (experience trust)	*p-value*
0	59.38	59.38	=1.00
50	61.48	61.57	>0.67
100	60.23	60.02	>0.30
150	60.22	60.33	>0.60
200	60.21	60.36	>0.45

In summary, in both cases of sharing the combination trust and sharing only the experience trust, there is no significant difference of average quality of products of all buyers. Furthermore, the results are also not depended on either the number of bought products or the number of friends of each buyer agent. Thus, the experimental results suggest that we can apply any one of these two strategies of sharing the trust in the applications of multi-agent system in e-commerce. The strategy of sharing the single experience trust is more effective since it could make lower cost of computation and an trust-based architecture simpler than the combination trust.

5. **Discussions.** Our computational trust model is one hybrid solution for an agent's decision making to choose partners in coordination. As presented in Section 1, the current hybrid models have some drawbacks due to using propagation mechanisms and weighted integration technique of reference trusts. This section is devoted to discussing how our model deals with such limitation.

Consider a scenario in which agent i has two partners j and l. i has a hight personal trust (experience trust) on j but the community has a low trust (reputation trust) on j. In contrast, for l, i has a low personal trust on l but the community has a hight trust on l. Which partner will agent i choose to interact? A weighted averaging of personal and community trusts as proposed this model will help agent i to choose the most trustworthy partner.

5.1. **Avoiding of redundant and recursive reference.** The redundant reference occurs when agents share the final trust about third partner instead of experience trust to his friends. For example, Figure 7 (left side) illustrates the case, each circle represents an overall trust which is composed of experience trust and reference trust, each arrow represents the reference among agent about reference trust.

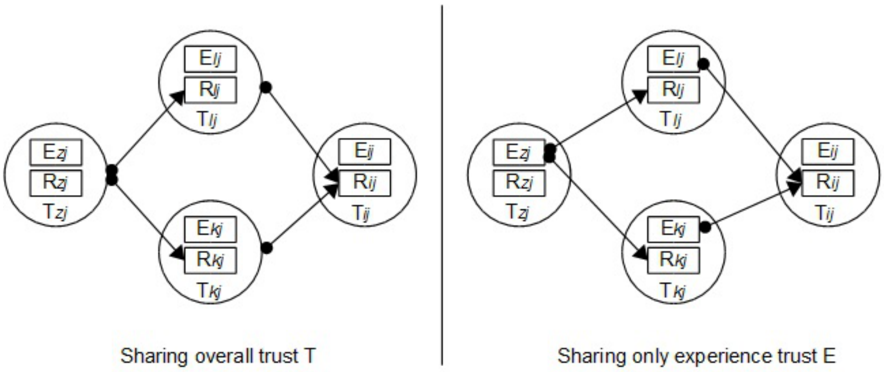


FIGURE 7. Mechanism of reference: sharing of overall trust or only experience trust

Assume that agent l sends to agent i her/his overall trust about agent j. Then T_{lj} is composed of experience trust E_{lj} and reference trust R_{lj} of l on j, where R_{lj} is calculated from agent z's overall trust T_{zj} on j. Likely, agent k also sends to i her/his overall trust about j. Then T_{kj} is also composed of E_{kj} and R_{kj} and R_{kj} is also calculated from z's overall trust T_{zj} on j. Reference trust R_{ij} is then calculated from T_{lj} and T_{kj}. It means that R_{ij}, which is affected by T_{lj} and T_{kj}, will be affected at least two times by T_{zj}. The computation of reference trust value may result in explosion when more agents need to evaluate more partners.

Likely, the recursive reference will occur when, for instance, agent i and k refer with each other (about agent j). Suppose R_{ij} refers from T_{kj} which is composed of E_{kj} and R_{kj}; and R_{kj} refers from T_{ij} which is composed of E_{ij} and R_{ij}, and so on. R_{ij} thus refers recursively to itself many times.

Our model may avoid these problems by sharing only the personal trust E_{ij} instead of overall trust T_{ij} in reputation's mechanism (Figure 7, right side, for the case of avoiding of redundant reference). R_{ij} is influenced by the experience trust E_{kj} and E_{lj} which are independent from T_{zj}. There are only reference trust R_{kj} and R_{lj} depending on experience trust E_{zj}, but they have no effect on reference trust R_{ij} of agent i about agent j.

The results of experiment 2 already indicated that there is no significant difference in results of two strategies: sharing the overall trust and sharing only experience trust. This leads to the fact that the strategy of sharing single experience trust makes the low cost in computation and the trust architecture simpler compared with the strategy of sharing the overall trust.

5.2. **Average aggregation of reference.** In our hybrid computational trust model, the most important characteristics of our reference mechanism is average calculation: The reference trust is an average aggregation of experience trusts from all referenced agents. It means that our model does not assign the different weights for agent's friends who give him/her their experience trust about a third partner. And non-weight average computation significantly reduces the cost of computation in our model.

A question is that whether or not it is necessary to assign the different weights to agent's friends in sharing reference trust. Suppose the answer is positive. Then, there are two possibilities for constructing the weights of agent's friends in sharing reference trust. First, we would use experience trust E_{ij} as the weight of reference trust referenced from agent j. This is not reasonable because E_{ij} represents the trust in all transactions between i and j, which is naturally different from the trust in sharing information between such two agents. Moreover, if it were the case, the experience trust E_{ij} would affect two times in overall trust: one time directly as definition of overall trust, one indirectly from reference trust R_{ij}. Such construction of weighted reference trust is not reasonable.

Second, we would construct a new system for trust sharing which may be the same principle with the estimate way of experience trust, but completely independent from it. Thus, an agent would need to judge the trust for each time a friend shares a reference trust with her/him. Then managing all the share times in the past and generating a vector of weights for the aggregation may result in complexity for our estimation.

These reasons have supported our thinking in taking the average aggregation of all referenced agent's experience trust rather than assigning weights to them.

6. **Conclusions and Future Work.** This paper has introduced a computational trust model to enable agents to calculate, estimate and update the degree of trust on their partners based both on their own experiences, and the judgment of agent society about the trustworthiness of partners. We have investigated our model from various aspects with experiments in an e-commerce environment. The experimental results indicate that it is better to use the combination of the experience and reference trusts for estimating the trustworthiness of a partner. Furthermore, the results suggest that we can apply any one of two strategies to share information about trust in multi-agent system: Either sharing the final combination trust or sharing the single experience trust. And it is clear that using the single experience trust in sharing among agent community may reduce the cost in computation and help construct the trust-based system architecture simpler.

Our model, as well as most computational trust models, has been constructed with the assumption that all agents are truthful. However, selfish or malicious agents may dispatch false information or make use of unreliable resources. The problem is how to discover or to restrict impacts of false or unreliable information resources in trust computation. We are currently considering an extension of our model to deal with these issues. The research results will be presented in our future work.

REFERENCES

[1] J. Carter, E. Bitting and A. Ghorbani, Reputation formalization for an information-sharing multi-agent system, *Computational Intelligence*, vol.18, no.2, pp.515-534, 2002.

[2] D. Gambetta, Can we trust trust? in *Trust: Making and Breaking Cooperative Relations*, D. Gambetta (ed.), New York, Basil Blackwell, 1990.

[3] T. Grandison and M. Sloman, A survey of trust in internet application, *IEEE Communications Surveys, Fourth Quarter*, 2000.

[4] T. D. Huynh, N. R. Jennings and N. R. Shadbolt, An integrated trust and reputation model for open multi-agent systems, *Autonomous Agents and Multi-Agent Systems*, vol.13, no.2, pp.119-154, 2006.

[5] A. Josang, C. Keser and T. Dimitrakos, Can we manage trust? *Proc. of the 3rd International Conference on Trust Management (iTrust)*, Paris, 2005.

[6] X. Li and X. Gui, Tree-trust: A novel and scalable P2P reputation model based on human cognitive psychology, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3797-3807, 2009.

[7] D. W. Manchala, E-commerce trust metrics and models, *IEEE Internet Comp.*, pp.36-44, 2000.

[8] S. Nefti, F. Meziane and K. Kasiran, A fuzzy trust model for e-commerce, *Proc. of the 7th IEEE International Conference on E-Commerce Technology (CEC'05)*, pp.401-404, 2005.

[9] S. D. Ramchurn, C. Sierra, L. Godo and N. R. Jennings, Devising a trust model for multi-agent interactions using confidence and reputation, *International Journal of Applied Artificial Intelligence*, vol.18, no.9-10, pp.833-852, 2004.

[10] J. Sabater and C. Sierra, Reputation and social network analysis in multi-agent systems, *Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, pp.475-482,2002.

[11] M. Schillo, P. Funk and M. Rovatsos, Using trust for detecting deceitful agents in artificial societies, *Applied Artificial Intelligence (Special Issue on Trust, Deception and Fraud in Agent Societies)*, 2000.

[12] S. Sen and N. Sajja, Robustness of reputation-based trust: Booblean case, *Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, pp.288-293, Bologna, Italy, 2002.

[13] J. Shibata, K. Okuhara, S. Shiode and H. Ishii, Application of confidence level based on agent's experience to improve internal model. *International Journal of Innovative Computing, Information and Control*, vol.4, no.5, pp.1161-1168, 2008.

[14] P. Victor, C. Cornelis, M. De Cock and P. P. da Silva, Gradual trust and distrust in recommender systems, *Fuzzy Sets and Systems*, vol.160, no.10, pp.1367-1382, 2009.

[15] R. R. Yager, Nonmonotonic OWA operators, *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, vol.3, no.3, pp.187-196, 1999.

[16] B. Yu and M. P. Singh, Distributed reputation management for electronic commerce, *Computational Intelligence*, vol.18, no.4, pp.535-549, 2002.

[17] L. A. Zadeh, A computational approach to fuzzy quantifiers in natural languages, *Computers and Mathematics with Applications*, pp.149-184, 1983.