CBTM: A Trust Model with Uncertainty Quantification and Reasoning for Pervasive Computing

Rui He, Jianwei Niu, and Guangwei Zhang

Beijing University of Aeronautics and Astronautics,
Xueyuan Road 37, Beijing, China, 100083
{harry.he, niujianwei}@buaa.edu.cn, ezhang@263.net

Abstract. This paper presents a novel trust model in which we model trust based on an exotic uncertainty theory, namely cloud model. We regard trust between entities as a cloud that is called as trust cloud. Based on such a quantification model of trust, we further propose the algorithms to compute propagated trust relationships and aggregated trust relationships, which are needed for trust reasoning in pervasive computing environments. Finally, we compare the proposed trust model with other three typical models in simulation experiments, and the results shows the cloud-based trust model performs better in a total sense.

1 Introduction

Trust has been researched for more than teen years since Marsh's work [1]. Because trust mechanism is more flexible and extensible than traditional security approaches such as PKI[20], trust has been introduced into many other cyber fields, i.e. pervasive computing, peer-to-peer networks, etc. In such contexts, trust is always regarded as subjective, therefore, how to measure trust become very important. Till now many approaches have been proposed to quantify trust [1,2,3], which either use discrete numbers such as *-1, 0, 1*, etc. to indicate different trust levels, or use a real number interval, for instance *[0, 1]*. However, since trust is subjective, it is not enough to describe trust with deterministic values.

As we know, in human society, when we say we trust a person *very much*, actually we are not so sure about to what an accurate degree to trust him or her. On the other hand, we can trust two persons both *very much,* but we may trust one a little more than the other. The same can be applied to pervasive computing environments. Hence we declare that uncertainty is an important nature of trust, which means trust relationships between entities are fuzzy and stochastic. For example, for two completely unacquainted entities, they may trust each other to *a little* degree, so that they can begin to cooperate in a task. Meanwhile, two familiar entities can also trust one another to *a little* degree, which may result from their bad interaction history. From these two cases, we can see that, regarding the same trust description, say trust *a little*, the former is absolutely uncertain but the latter is quite assure. Therefore, we must incorporate uncertainty when modeling trust.

In this paper, we propose such an trust model, namely the cloud based trust model or CBTM. We will present an overview of the cloud model in section 2. And in section 3 we will delineate the cloud based trust model in detail. Then simulation ex-

Y. Pan et al. (Eds.): ISPA 2005, LNCS 3758, pp. 541 – 552, 2005.
© Springer-Verlag Berlin Heidelberg 2005

periments will be presented in section 4. Related work will be listed in section 5. Finally, we will summarize our work and point out our future work in section 6.

2 Cloud Theory Overview

The cloud model was firstly proposed as a model of the uncertainty transition between a linguistic term of a qualitative concept and its numerical representation [9]. Till now, the cloud model has been applied in many fields successfully, such as automatic control [11], knowledge discovery and data mining [10,12], etc

Formally, a cloud can be defined as follows [22].

DEFINITION 1: Let U be the set as the universe of discourse, f is a random function with a stable tendency $f : \mathrm{U} \to [0,1]$, and g is also a random function with a stable tendency $g : \mathrm{U} \to \mathrm{U}$, He is an uncertain factor and $0 \le He$, and

1) $u' = g(u, He), u \in \mathrm{U}$.
2) $y = f(u', He)$.

Then (U, g, f, He) is a cloud, and (u', y) is a cloud drop.

In DEFINITION 1, the mapping f from U to the interval [0,1] is a one-point to multi-point transition, so the degree of membership of u is a probability distribution rather than a fixed value, which is the very place where the cloud theory is different from the fuzzy logic. For example, a one-dimension normal cloud can be formalized as follows.

$$\begin{aligned} nc &= (\mathbf{R}, g, f, He) \\ g &= \mathrm{randn}(Ex, \mathrm{randn}(d, He)) \\ x' &= g(x, He) \\ f &= e^{-\frac{(x'-Ex)^2}{2\times \mathrm{randn}(d,He)^2}} \end{aligned} \quad . \tag{1}$$

where $d, x \in \mathbf{R}$ ($\mathbf{R}$ is the set of real numbers) and randn(a, b) is a normally distributed random number generation function with a as the mean and b as the standard deviation.

For the purpose of simpliness, normal clouds defined by Formula (1) can be denoted by three digital characteristics [9], namely Expected Value (*Ex*), Entropy (*En*) and Hyper-Entropy (*He*). With these digital characteristics, the fuzziness and randomness of uncertain concepts can be integrated in a unified way. The expected value *Ex* points out the center of gravity of a normal cloud. The entropy *En* is a measure of the fuzziness of the concept over the universe of discourse. It shows the span of cloud drops distribution. The hyper entropy *He* is a measure of the uncertainty of the entropy *En*. And the greater *He* is, the more dispersedly the membership degrees are distributed. In the extreme case, both entropy *En* and hyper entropy *He* is equal to zero, namely (*Ex, 0, 0*), which presents the concept of a deterministic datum.

It is easy to see that the *He* in Formula (1) is the same as the *He* in DEFINITION 1. Furthermore, based on the normal cloud definition, and given three digital characteristics, say *Ex, En, He*, we can build a normal cloud with the so-called normal cloud generator, which is described by the following algorithm [9].

ALGORITHM 1: Given a normal cloud (*Ex, En, He*) and the number of the cloud drops *N*, a normal cloud can be computed following steps as follows.

1) Produce a normally distributed random number *En'* with the mean *En* and the standard deviation *He*;

2) Produce a normally distributed random number *x* with the mean *Ex* and the standard deviation *En';*

3) Calculate $y = e^{-\frac{(x-Ex)^2}{2(En')^2}}$;

4) Point (*x, y*) is a cloud drop in the universe of discourse;

5) Repeat steps 1-4 until *N* cloud drops are generated.

Fig. 1 illustrates the normal cloud description of the term "10 miles around".

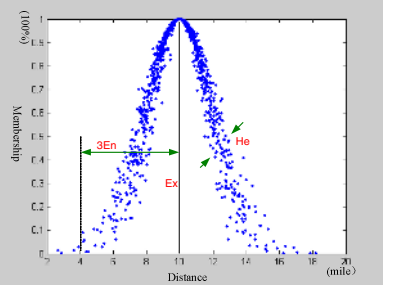


Fig. 1. Cloud shape and three digital characteristics of the linguistic term "10 miles around"

3 Cloud-Based Trust Model

Based on the cloud model, we research on uncertainty of trust and propose a novel cloud based trust model or CBTM, which will be described in this section in detail.

3.1 Trust Cloud

The trust cloud is the core concept of CBTM, which is defined as follows.

DEFINITION 2: A trust cloud is a normal cloud to quantify a trust relationship between two entities, indicating how much and how surely one is trusted by the other. Formally, the trust cloud held by an entity, i.e. *A*, about the other entity, i.e. *B*, can be denoted as:

$$\begin{aligned} &tc_{AB} = nc(Ex, En, He) \\ &0 \le Ex \le 1 \\ &0 \le En \le 1 \\ &0 \le He \le 1 \end{aligned} \quad . \tag{2}$$

where *nc(Ex, En, He)* is a normal cloud defined by Formula (1), and *Ex* is the trust expected value here, which indicates the basic trust degree of *B* for *A*. *En* reflects the uncertainty of the trust relationship. It also describes the scope of cloud drops which can be accepted by *A*, namely the fuzziness degree. And *En* shows the stochastic density of the cloud drops in the trust space, namely the randomness of the trust relationship. *He* is the trust hyper entropy here, which indicates the uncertainty of fuzziness degree of the trust relationship.

It should be pointed out that, when $En \neq 0$ and $He = 0$, the trust relationship of entity *A* to entity *B* is fuzzy, but the fuzziness degree is deterministic; and when $En = 0$ and $He = 0$, the trust relationship of entity *A* to entity *B* is deterministic and there is no uncertainty in the trust. For example, the entities belonging to an internal system or the same administrative domain could have deterministic trust relationships.

Fig. 2 illustrates some typical trust clouds.

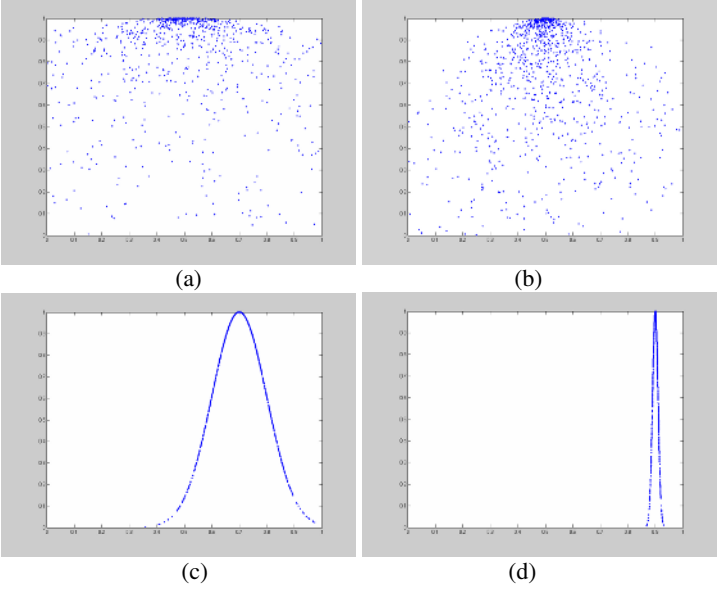


Fig. 2. Typical trust clouds. (a) tc (0.5, 0.1, 0.6). (b) tc (0.5,0.1,0.2). (c) tc (0.7,0.1,0). (d) tc(0.9,0.01,0).

From Fig.2 we can deduce that: (1) The greater *Ex* is, the closer a trust cloud approaches to the max trust value, namely 1; (2) The greater *En* is, the wider the span of a trust cloud is; (3) The greater *He* is the more dispersive the cloud drops of a trust cloud are.

3.2 Distrust and No trust

In trust modeling, distrust and no trust have different meanings. If entity *A* distrusts entity *B*, it means *A* knows *B* can not be trusted. On the contrary, if entity *A* has no trust about entity *B*, it means *A* does not know whether or how much *B* should be

trusted. Traditionally, different values are used to distinguish distrust and no trust. For example, -1 indicates no trust, and 0 indicates distrust [13]. However, we declare that distrust and no trust are two different concepts describing trust from different viewpoints, namely trustworthiness viewpoint and uncertainty viewpoint. Therefore, more should be done rather than just assigning different values to them.

From the standpoint of the cloud model, distrust is used to describe trust relationships from the aspect of trustworthiness degree, and we can denote distrust with $Ex = 0$. And no trust is a concept describing trust relationships from the aspect of uncertainty of trust, and it can be indicated by setting $En = 1$ and $He = 1$. Therefore, we can see that distrust and no trust are two intercrossed concepts and in some cases both of them can even co-exist in one trust relationship. For example, entity A meets a stranger entity B, and then the trust relationship A established to B should be no trust. At the same time, A may also label the trust degree as distrust, because A is very cautious. On the contrary, if A is adventurous, A may set an average trust degree to the no trust relationship. In a word, distrust and no trust are not exclusive. This distinguishes CBTM from all other trust models.

Fig. 3 shows some examples of distrust clouds and no trust clouds.

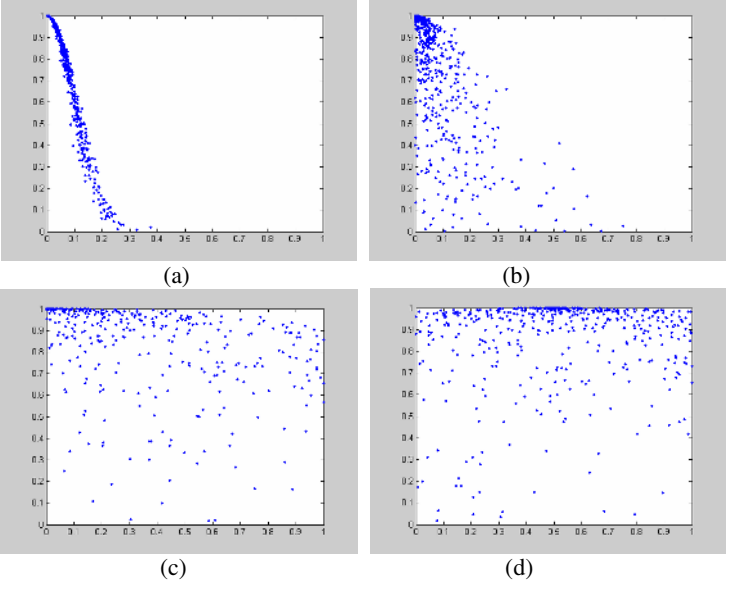


Fig. 3. Distrust and unknown trust clouds. (a) distrust cloud tc(0,0.1,0.01). (b) distrust cloud tc(0,0.1,0.1). (c) no trust and distrust cloud tc(0,1,1). (d) no trust cloud tc(0.5,1,1).

3.3 Trust Reasoning

In pervasive computing environments, unknown entities are always met. Before these strangers cooperate with each other, their trustworthiness should be determined. However, as strangers, their trust degrees can not be known by one another at present.

Therefore, it is necessary for any entity to derive a trust relationship to the stranger based on exiting trust relationships. Therefore algorithms of computing trust clouds are needed for trust reasoning.

In CBTM, trust cloud computation consists of two parts, namely computing a new trust cloud through trust propagation and combining many trust clouds into one unique trust cloud.

Propagating Trust Clouds

In pervasive computing environments, e.g. ad-hoc networks, entities always can not get trust recommendation of a stranger from their trusted neighbors directly, so trust cloud propagation is needed.

Supposing there are m entities, say A_1, A_2, A_3, …, A_m, and the trust cloud from A_i to A_{i+1} $(1 \le i \le m\text{-}1)$ is $tc_i(Ex_i, En_i, He_i)$, then the trust cloud of A_1 about A_m, denoted as $tc(Ex, En, He)$, can be computed as follows.

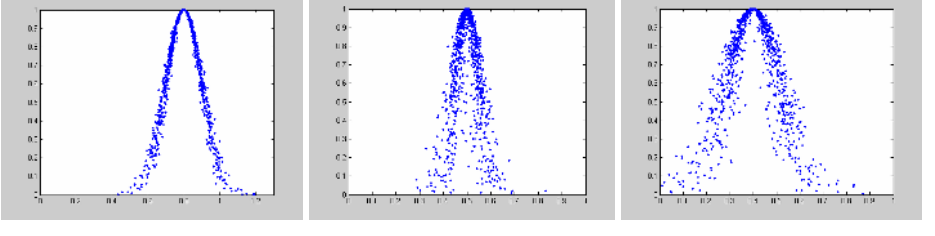
$$tc(Ex, En, He) = tc_1 \otimes tc_2 \otimes \ldots \otimes tc_m = \prod_{i=1}^{m} tc_i(Ex_i, En_i, He_i)$$

$$\begin{aligned} Ex &= \prod_{i=1}^{m} Ex_i \\ En &= \min\left(\sqrt{\sum_{i=1}^{m} En_i^2}, 1\right) \\ He &= \min\left(\sum_{i=1}^{m} He_i, 1\right) \end{aligned} \quad . \tag{3}$$

Where $\otimes$ is called cloud logic multiplicative operator.

For instance, suppose A trusts B as $tc_{AB}(0.8, 0.1, 0.01)$, and B trusts C as $tc_{BC}(0.5, 0.05, 0.02)$, then the trust cloud held by A to C, denoted as $tc_{AC}(Ex, En, He)$, can be computed according to Formula (3) as follows.

$$\begin{aligned} Ex &= 0.8 \times 0.5 = 0.4 \\ En &= \min\left(\sqrt{0.1^2 + 0.05^2}, 1\right) \approx 0.112 \\ He &= \min(0.01 + 0.02, 1) = 0.03 \end{aligned} .$$



(a) (b) (c)

Fig. 4. Propagating trust cloud. (a) $tc_{AB}(0.8, 0.1, 0.01)$. (b) $tc_{BC}(0.5, 0.05, 0.02)$.(c) tc_{AC} (0.4, 0.112, 0.03).

We illustrate these three trust clouds in Fig. 4, from which it is easy to see that after propagation, the trust cloud becomes more dispersive and closer to 0. This means the trust degree is decreased and the uncertainty is increased. This accords with human experience.

Aggregating Trust Clouds

In many cases, more than one trust clouds of a stranger entity can be computed, therefore, it is necessary for an entity to combine these trust clouds into a unique one.

Supposing there are m trust clouds, say tc_1, tc_2, tc_3, …, tc_m, then these trust clouds can be combined into one trust cloud, say tc(*Ex, En, He*), as follows.

$$
\begin{aligned}
tc(Ex, En, He) &= tc_1 \oplus tc_2 \oplus \ldots \oplus tc_m = \sum_{i=1}^{m} nc_i(Ex_i, En_i, He_i) \\
Ex &= \frac{1}{m}\sum_{i=1}^{m} Ex_i \\
En &= \min(\frac{1}{m}\sum_{i=1}^{m} En_i, 1) \\
He &= \min(\frac{1}{m}\sum_{i=1}^{m} He_i, 1)
\end{aligned}
\tag{4}
$$

Where $\oplus$ is the cloud logic additive operator.

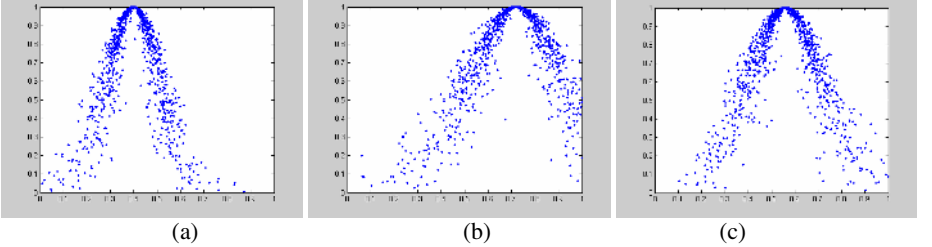
For example, entity A gets two propagated trust clouds, i.e. $tc_1(0.4, 0.112, 0.03)$ and $tc_2(0.72, 0.2, 0.05)$, then the aggregated trust cloud tc(*Ex*, *En*, *He*) can be computed according to Formula (4) like this.

$$Ex = (0.4 + 0.72)/2 = 0.56$$

$$En = \min((0.112 + 0.2)/2, 1) = 0.156$$

$$He = \min((0.03 + 0.05)/2, 1) = 0.04$$

These three trust clouds are illustrated in Fig.5, from which we can see the combined trust cloud is between the two operand trust clouds from both aspect of trust level and uncertainty. This also accords with our intuition.



(a) (b) (c)

Fig. 5. Aggregated trust clouds. (a)tc1(0.4,0.112,0.03) (b)tc_2(0.72,0.2,0.05) (c)tc (0.56,0.156, 0.04).

4 Simulation Experiment

Since using cloud to model trust is absolutely exotic, it is necessary for us to prove the validation of CBTM in experiments. Therefore, we carried out a simulation experiment.

Our experiment is based upon a simulation platform RePast[19],which is popular in simulating agent-based systems. Over RePast, we developed CBTM. As comparison, we also implemented other three trust models, which are based on Depster-Shafer theory, probability theory, and Fuzzy logic respectively. These trust models are:

- Yu Trust Model (abbreviated as Y Model) [18]
- Beth Trust Model (abbreviated as B Model) [2]
- Tang Trust Model (abbreviated as T Model) [21]

In the experiment, our proposed trust model is abbreviated as C Model.

4.1 Metrics

To compare selected trust models quantitatively, we define some metrics first.

DEFINITION 3: Suppose $\mathbf{E} = \{A_i \mid 1 \le i \le N\}$ are the set of entities in a pervasive environment, and $\mathbf{Tr} = \{(A_i, A_j, tval_{ij}) \mid A_i, A_j \in \mathbf{E}, 0 \le tval_{ij} \le 1\}$ is the set of trust relationships between these entities, then we define average trust density (ATD) as

$$\mathrm{ATD} = \frac{\sum_{tval_{ij} \in \mathbf{Tr}} tval_{ij}}{\mathrm{P}_N^2} = \frac{\sum_{tval_{ij} \in \mathbf{Tr}} tval_{ij}}{N \times (N-1)}. \tag{5}$$

This metric represents the overall trust level of a network. If the ATD of a network is too low, it means the society formed by the network is fragile and it is easy to collapse. At the same time, the faster the ATD curve become horizontal, the better a trust model's convergence is.

DEFINITION 4: Suppose the total interaction (from service request to its being permitted or denied) number between entities in the network is N_t, and total successful cooperation (service request is permitted) number is N_s, then we define successful cooperation probability (SCP) as

$$\mathrm{SCP} = \frac{N_s}{N_t} \times 100\% . \tag{6}$$

This metric shows the cooperation level of a network. The greater this metric is, the more cooperative the society and a trust model are.

DEFINITION 5: Suppose the time an entity receives a request is t, and the time a trust model finishes evaluating the requester entity's trust is t', and the total number of interaction in the network is N, then we define average response delay (ARD) as

$$\mathrm{ARD} = \frac{\sum_{i=1}^{N} (t_i - t_i^{'})}{N} \tag{7}$$

This metric shows the complexity of a trust model. Since our simulation does not consider physical network delay, the delay time is due to trust model computation. So the bigger ARD is the more complex a trust model is. And the less complex a trust model is, the better it is.

4.2 Simulation Parameter Setting

In the experiment, we created a network with specific number of entities and the entities in it are reachable for one another. During initialization, each entity was assigned randomly the specific number of acquaintances, and the trust relationships between them were initialized randomly.

During the experiment, entities interacted with each other for specific times. In each interaction, the simulation system chose two entities randomly, and the first was requester and the other was server. The server computed the requester's trustworthiness using a trust model, and decided whether the request would be accepted not by comparing the evaluation result with the predefined cooperation threshold value. In each interaction, every trust model was used and concerned data were recorded.

The simulation system parameter setting is described in Table 1.

Table 1. Simulation system parameter setting

Parameter	Value
Initial Acquaintance.	5
Entity Number	100
Interaction Number	25×2500
Threshold Value	0.5

4.3 Experiment Results

The experiment results are illustrated in Fig. 9.

From Fig.9 (a), we can see that the proposed C model and Y model's NTD are very close and much higher than both B model and T model. But C model becomes convergent faster than Y model.

From Fig.9 (b), we can observe that our proposed C model has a much far better performance than all the other models in terms of successful cooperation probability. This indicates CBTM will provide entities more chances to cooperate with each other.

From Fig.9 (c), we can see that the ARD of C model is much lower than the other models, which indicates CBTM is much easier and will consume less CPU time.

Based on all the experiment results, we can tell that CBTM performs quite well in terms of convergence, cooperation, and complexity.

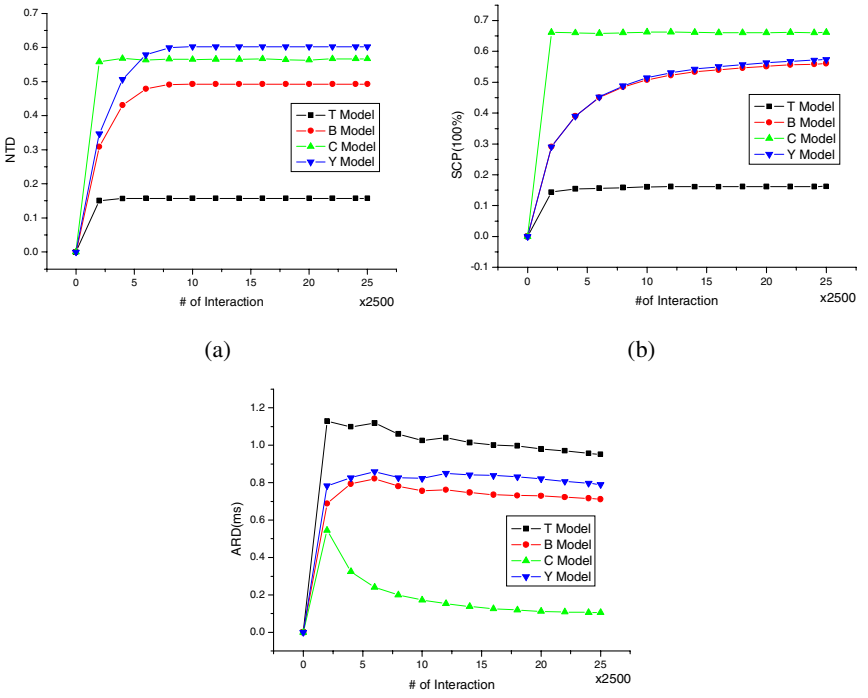


Fig. 9. Experiment results. (a) NTD (b) SCP (c) ARD.

5 Related Work

In the computer science literature, Marsh is among the first to study trust. In [1], he provided a clarification of trust and presented an implementable formalism for trust, and he applied his trust model in the distributed artificial intelligence (DAI) community to enable the agent to make trust-based decisions. Since his model attempted to integrate all the aspects of trust taken from sociology and psychology, it is rather complex.

At almost the same time, Beth et al. [2] also proposed a trust model for distributed networks. They considered trust in different classes, which are Per Se different functionalities in authentication protocols. Furthermore, they distinguished recommendation trust from direct trust and gave their formal representations, as well as rules to derive trust relationships and algorithms to compute trust values.

Another important trust model is proposed by Abdul-Rahman et al. [13]. They tried to give a model of generalized trust to be suited to trust relationships that are less formal, temporary or short-term. For this purpose, they classified trust relationships into two types, namely direct trust relationship and recommender trust relationship, which is quite different from recommendation trust in the model of Beth described above. Besides, they proposed a recommendation distribution protocol, as well as an algorithm to compute trust value of target for a single recommendation path.

Following these basic work, a lot of trust models [4, 5, 16, 6, 15, 14,17] were proposed to various systems, including multi-agent systems, peer-to-peer networks, as well as pervasive computing. Unfortunately, these models do not consider uncertainty of trust at all. C. Castelfranchi et al [7], H. Zhuang et al [8], and Tang [22] did consider uncertainty, more accurately, fuzziness, and they used fuzzy logic to deal with trust related problems. It is their work to inspire us to research the uncertainty of trust deeply.

6 Conclusion and Future Work

In this paper, we propose a novel trust model, namely the cloud based trust model or CBTM. Distinguished from previous trust models, CBTM takes uncertain of trust into account and describes the trust degree and trust uncertainty in a uniform form, namely cloud. In CBTM, we give the cloud description of trust as well as algorithms to compute propagated trust values and aggregated trust values. And our simulation experiment demonstrates the better performance of CBTM preliminarily.

As for our future work, we will continue to perfect CBTM. We will incorporate other factors into current model, such as risk, reputation, etc. In other words, we will work on a more complex model, which will be more practical to deal with trust issues in pervasive computing. Besides, we will consider cheating or vicious behaviors in pervasive computing environments and methods will be researched to detect such behaviors, and further reduce or even prevent them.

References

[1] S. P. Marsh. Formalising Trust as a Computational Concept. Ph.D. Thesis, University of Stirling, 1994.

[2] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In ESORICS 94. Brighton, UK, November 1994

[4] Bin Yu, Munindar P. Singh, An Evidential Model of Distributed Reputation Management, Proceedings of First International Joint Conference on Autonomous Agents sand Multi-Agent Systems, pages 294-301, 2002

[5] K. Aberer, Z. Despotovic. Managing Trust in a Peer-2-Peer Information System. In Proceedings of the 10th International Conference on Information and Knowledge Management (ACM CIKM), New York, USA, 2001.

[6] Huafei Zhu, Bao Feng, Robert H. Deng. Computing of Trust in Distributed Networks. http://venona.antioffline.com/2003/056.pdf

[7] Cristiano Castelfranchi, Rino Falcone, Giovanni Pezzulo. “Trust in Information Sources as a Source for Trust: A Fuzzy Approach”. In Proceedings of the second international joint conference on Autonomous agents and multiagent systems, July 14-18, 2003. pp. 89-96

[8] Hanqi Zhuang, Songwut Wongsoontorn, Yuanhui Zhao. A Fuzzy-Logic Based Trust Model and its Optimization for e-Commerce. F Florida Conference on the Recent Advances in Robotics (FCRAR 2003).

[9] Deyi LI, Haijun MENG, Xuemei SHI. Membership clouds and membership clouds generator. Journal of Computer Research and Development, 42(8): 32-41, 1995.

[10] Deren Li, Shuliang Wang, Wenzhong Shi, Xinzhou Wang, 2001, On spatial data mining and knowledge discovery (SDMKD), Geomatics and Information Science of Wuhan University, 26(6):491-499
[11] Deyi Li. The Cloud Control Method and Balancing Patterns of Triple Link Inverted Pendulum Systems. Chinese Engineering Science. Vol 1, No 2, p41-46, Nov 1999.
[12] Deyi Li. Uncertaintyin Knowledge Representation. Chinese Engineering Science. Vol 2, No 10, p73-79, Oct 2000.
[13] A. Abdul-Rahman and S. Hailes. A Distributed Trust Model. New Security Paradigms Workshop 1997, ACM, 1997.
[14] Brian Shand, et al. Trust for Ubiquitous, Transparent Collaboration. IEEE Pervasive Computing and Communication 2003.
[15] L. Mui, M. Mohtashemi, A. Halberstadt. "A Computational Model of Trust and Reputation," 35th Hawaii International Conference on System Science (HICSS), 2002.
[16] R. Chen and W. Yeager, "Poblano: A Distributed Trust Model for Peer-to-Peer Networks", Sun Microsystems Technical Paper, 2000, http://www.sun.com/software/jxta/poblano.pdf
[17] M. Carbone, M. Nielsen and V. Sassone. A Formal Model for Trust in Dynamic Networks. Proceedings of IEEE International Conference on Software Engineering and Formal Methods (SEFM '03), 2003.
[18] Bin Yu, Munindar P. Singh, An Evidential Model of Distributed Reputation Management, Proceedings of First International Joint Conference on Autonomous Entities and Multi-Entity Systems, pages 294-301, 2002
[19] RePast WebSite: http://repast.sourceforge.net/
[20] IETF PKIX Working Group. http://www.ietf.org/html.charters/pkix-charter.html
[21] Wen Tang. The Research on Fuzzy Set Theory Based Trust Management. Beijing University. PhD Thesis. 2003.
[22] Deyi Li, Kaichang Di, Deren Li, and Xuemei Shi, Mining Association Rules with Linguistic Cloud Models, Proceedings of PAKDD98, Australia, 15-17 April, 1998, Springer-Verlag Heidelberg, P392-394.