CertainTrust: A Trust Model For Users And Agents

Sebastian Ries*
Department of Computer Science
Darmstadt University of Technology
Hochschulstrasse 10
64289 Darmstadt, Germany
ries@tk.informatik.tu-darmstadt.de

ABSTRACT

One of the challenges for ubiquitous computing and P2P systems is to find reliable partners for interactions. We believe that this problem can be solved by assigning trust values to entities and allowing them to state opinions about the trustworthiness of others. In this paper, we develop a new trust model, called CertainTrust, which can easily be interpreted and adjusted by users and software agents. A key feature of CertainTrust is that it is capable of expressing the certainty of a trust opinion, depending on the context of use. We show how the trust values can be expressed using different representations (one for users and one for software agents) and present an automatic mapping to change between the representations.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Models—*Human-centered computing*; I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence

General Terms

Design, Security, Theory

Keywords

trust model, evidence, recommendations, user interface

1. INTRODUCTION

In [1], Bhargava et al. point out that "trust [...] is pervasive in social systems" and that "socially based paradigms will play a big role in pervasive-computing environments". Pervasive or ubiquitous computing is characterized by a very large number of smart devices, e.g., PDAs, mobiles, intelligent clothes, etc., which come with different capabilities considering communication channels, storage or battery power.

Both, the basic idea of ubiquitous computing and the heterogeneity of these devices enforce interaction with and delegation to other devices to enfold the complete potential of an ubiquitous computing infrastructure. Ubiquitous computing environments are unstructured and many service providers are only locally or spontaneously available. Therefore, we cannot expect to know for sure, if those devices will behave as expected and cooperate, or not.

On the one hand, the interactions with devices, which are not possessed or controlled by ourselves, include uncertainty and risk, since a safe prediction of the behavior of those devices is not possible. On the other hand, the interactions with reliable partners are the basis for the services ubiquitous computing environments can provide. But how to select reliable interaction partners and delegatees who behave as expected? Selecting only tamper-proof devices, which belong to the same manufacturer, requires the manufactures to be trusted, and unnecessarily reduces the potential of ubiquitous computing.

Due to the great number of interactions with many different partners – some might be well-known, others not – and the claim of ubiquitous computing to become a calm technology, we need a non-intrusive way to cope with this challenge.

We believe that the concept of trust, which has proofed to work well in real life, is a promising solution, which allows to make well-founded decisions even in the context of risk and uncertainty. Assuming recognition of entities, e.g. [14], trust allows to express an expectation about the future behavior of an entity based on evidence collected in past engagements. Since ubiquitous computing enforces a human-centric design, trust needs representations, which are meaningful not only to the software agents to enable automatic trust evaluation, but also to the end user, who needs to be able to reflect about the state of the trust model, and to take part in the decision making process, if necessary.

In this paper, we provide a decentralized trust model, named *CertainTrust*, which allows agents to choose trustworthy partners for risky engagements. For our trust model, we propose two representations. The first one serves as basis for a human trust interface. It allows to represent trust using two independent parameters, consisting of an estimate for the probability of trustworthy behavior in a future engagement, and of a parameter expressing the certainty of this estimate. Since we believe that trust is context-dependent,

*The author's work was supported by the German National Science Foundation (DFG) as part of the PhD program "Enabling Technologies for Electronic Commerce" at Darmstadt University of Technology

SAC'07 March 11-15, 2007, Seoul, Korea

we also enforce the context-dependency of the certainty parameter of a trust value. The second representation is based on the Bayesian approach, using beta probability density functions. This approach is well-established to express trust, and serves as a basis for the trust computation and as an interface for evidence-based feedback integration. Finally, we provide a mapping between both representations, and operators for 'consensus' and 'discounting'.

The remainder of this paper is structured as follows. In Section 2, we summarize our notion of trust. Section 3 presents the trust model and the operators for trust propagation. Section 4 presents the related work, and Section 5 summarizes our contribution and outlines aspects for our future work.

2. OUR NOTION OF TRUST

From our point of view, trust is the well-founded willingness for a potentially risky engagement. Trust can be based on personal experience, recommendations, and the reputation assigned to the partner involved in an engagement. We model trust as the subjective probability, that an entity behaves as expected based on the Bayesian approach as introduced in [7].

2.1 Properties of Trust

The following properties, which are regularly assigned to trust, are part of our model. Trust is subjective, which means the trust of an agent A in an agent C does not need to be the same as trust of any other agent B in C, since the behavior of C towards A is not necessarily equal to C's behavior towards B. Furthermore, we cannot expect the behavior of A towards C being the same as the behavior of C towards A, therefore trust needs to be modeled asymmetric. Trust is context-dependent. Obviously, there is a difference in trusting in another agent as provider of mp3-files or as provider of an online banking service. There is also a difference in trusting in someone as service provider or as recommendation provider. If A trusts B in the context of providing recommendations about a good service provider, e.g., for file-storing, this does not necessarily mean, that A trusts in B as a good peer to store files at, and vice versa. Trust is non-monotonic, i.e., experience can increase as well as decrease trust. Therefore, we need to model positive and negative evidence. Trust is not transitive in a mathematical sense, but the concept of recommendations is very important, since recommendations are necessary to introduce trust in agents with which no or only little direct experience is available. Moreover, we do not think of trust as finite resource, e.g., as done in flow-based approaches like EigenTrust [8]. It should be possible to increase trust in agents without decreasing trust in other agents.

2.2 Arguments for a Certainty Value

Similar to [5, 10, 12], we believe that it is necessary to express the (un-)certainty or reliability of an opinion. We also believe that the certainty of an opinion increases with the number of evidence, on which an opinion is based. Modeling the certainty of an opinion allows to provide information on how much evidence an opinion is based, and especially to state that there is not any evidence available. Furthermore, it is possible to express, that one opinion might be supported by more evidence than another one.

Why does certainty need to be context-dependent (perhaps even subjective)?

- In ubiquitous computing environments trust models can be used to automate decision making in many different applications. In the context of some applications, there might be a great number of interactions, other applications might be related to high risk, considering legal or financial implications. In these contexts, it seems reasonable, that the users want to collect a great number of evidence, before they would think about an opinion to be certain. If forced to make a decision about an engagement involving high risk, one might choose to reject the engagement, although there is positive but too little evidence.

- In contexts, in which the number of interactions is lower, or the associated risk is lower, the users may be satisfied with a lower number of evidence to come to a well-founded decision.

To model the context-dependency for the certainty of an opinion, we assume there is a *maximal number of expected evidence* per context, which corresponds to the maximal level of certainty. For example, the maximal number of expected evidence can be defined as 5, 10, 100, or 1000.

2.3 Trust vs. Reputation

Similar to [7], we see reputation as an opinion of the community about a single agent, whereas trust is the opinion of a single agent about another one. This goes along with the subjectivity of trust. This way, trust is not as dependant on the constitution of an agent society as reputation. That is, if in an agent society each agent has one vote, and there are more than 50% of agents in this society, which provide opinions stating that all other agents are bad, then all agents get a bad reputation. Whereas in a trust based system each agent is allowed to choose which opinions it wants to integrate in its calculated trust and how these opinions should be weighted. This way, each agent can judge the behavior of other agents only by its own opinion. It is not necessary for a trust metric to reflect about globally desirable behavior to produce reasonable trust scores. Since reputation can be a basis for trust [13], we allow the inclusion of reputation in our trust model. If reputation information is available, an agent can integrate the reputation in his calculation of trust in the same way as recommendations.

2.4 Scenario

As a scenario to show how trust can improve service provision in networks, we like to introduce a file-sharing scenario. Peers share files with others. If a peer or an agent has found another peer, who provides a file it wants to download, the peer has to decide whether to do it, or not. The risk of downloading a corrupted file depends on the assumption about the possible damage. If we assume a corrupted file does not lead to further damage, the risk is reduced to wasting bandwidth and CPU time. If we assume the file could contain viruses, the risk increases, since it can potentially damage our machine. If we use this machine for online-banking, or store personal data on it, the risk increases further.

The trust in a peer is based on direct experience, e.g., previous downloads, and recommendations from other trusted peers. Those recommendations are weighted by the trust in

the recommenders' ability to provide recommendations. After having calculated the trust value, and having estimated the potential risk, we can use this information for decision making. In this paper, we focus on calculating trust, but the involvement of risk and decision making is necessary to motivate the topic.

3. TRUST MODEL - CERTAINTRUST

Introducing the trust model we start presenting the general notation. Let contexts be denoted by con_i $i \in \{1, 2, \dots\}$, e.g., $con_1 = file\_sharing$ or $con_2 = online\_banking$. For providing recommendations for a context con_i, we define a special context, which is denoted as rec_i. Let agents be denoted by capital letters $A, B, \dots$. Let propositions be denoted by non-capital letters $x, y, \dots$. The opinion of agent A about the truth of a proposition x, e.g., x="Agent C behaves trustworthy in context con_i" (see Fig. 1), will be denoted as $o_x^A(con_i)$. The opinion of agent A about B's trustworthiness for providing recommendations for a context con_i will be denoted as $o_B^A(rec_i)$. If the context is non-ambiguous or non-relevant, we use o_x^A and o_B^A. The maximal number of expected evidence (see Section 2.2) is denoted as $e(con_i)$ or e. Since the evidence model is partly derived from ideas presented in [5], and to achieve better comparability, we use the same terminology when possible.

For the explanation of the trust model, we do not focus on trust management aspects as collecting and storing of evidence, risk assignment, and decision making. We assume that the evidence is collected and locally stored, and that recommendations are provided on request by the communication partners within range.

The propagation of recommendations is done based on chains of recommendations (see Fig. 1). We propose special operators for consensus (aggregation of opinions) and discounting (weighting of recommendations). For simplicity, opinions are assumed to be independent.

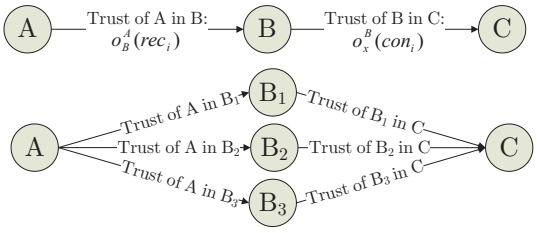

Figure 1: Trust chains

Our model provides two representations for opinions to express trust. The first representation is a pair of *trust value* and *certainty value* which serve as a base for a human trust interface. The second representation is based on the number of collected evidence and allows us to easily integrate feedback and forms the base of the computational model.

3.1 Human Trust Interface

The *human trust interface* (HTI) represents trust as opinions. In the HTI an opinion o is a 2-dimensional expression, represented by a 2-tuple $o = (t, c)^{HTI} \in [0, 1] \times [0, 1]$, where the superscript refers to the representational model. The opinion $o_B^A(rec_i) = (t_B^A(rec_i), c_B^A(rec_i))^{HTI}$ expresses the opinion of A about the trustworthiness of B in the context rec_i. The value of $t_B^A(rec_i)$ represents the probability with which A would consider the proposition "I believe, that B is trustworthy with respect to providing recommendations for the context con_i" to be true. This value is referred to as the *trust value*. The value $c_B^A(rec_i)$ is referred to as *certainty* or *certainty value*. This value expresses, which certainty the provider of an opinion assigns to the trust value. A low certainty value expresses that the trust value can easily change. Adversely, high certainty expresses the trust value is rather fixed. The values for trust and certainty can be assigned independently of each other. For example, an opinion $o_B^A(rec_i) = (1, 0.1)^{HTI}$ expresses that A expects B to be trustworthy in the context of providing recommendations for con_i, but this opinion can easily change, since A assigns a low certainty value. The interpretation for an opinion o_x^A is analogue.

For the moment, we express both the values for trust and certainty as continuous values in $[0, 1]$. Since humans are better in assigning discrete (verbal) values then continuous ones, as stated in [4, 7], we want to point out, that both values can easily be mapped to a discrete set of values, e.g., to the natural numbers in $[0, 10]$, or to set of labels, as e.g., *very untrusted* (vt), *untrusted* (u), *undecided* (ud), *trusted* (t), and *very trusted* (vt) for the trust value and *uninformed*, *rookie*, *average*, and *expert* for the certainty value.

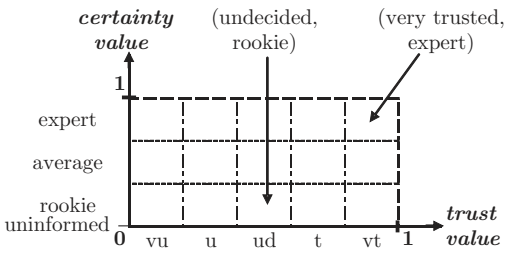

Figure 2: Example for a graphical HTI

To be able to introduce the semantics for the trust value and the certainty with labels, it is necessary that both parameters are independent of each other. Otherwise, the interpretation of the labels of the trust value would change with the certainty value, and vice versa. This would be counter-intuitive. If the both values are independent, the human trust interface allows the users to easily express and to interpret an opinion (see Fig. 2). This is important since it allows users to set up opinions. Furthermore, it allows the users to check the current state of the trust model, and to manually adjust an opinion, if they believe it is necessary.

3.2 Evidence Model

We now present the second representation, the evidence model which is based on beta probability density functions (pdf). The beta distribution $Beta(\alpha, \beta)$ can be used to model the posteriori probabilities of binary events. The corresponding pdf is defined by:

$$f(p \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \quad , \quad \text{where } 0 \le p \le 1,\ \alpha > 0,\ \beta > 0 \ . \tag{1}$$

Furthermore, we use $r = \alpha + 1$ and $s = \beta + 1$, where $r \ge 0$ and $s \ge 0$ represent the number of positive and negative evidence, respectively. The *number of collected evidence* is represented by $r + s$.

In the evidence model, an opinion o can be modeled using the parameters α and β. We refer to this representation with $o = (\alpha, \beta)^{\alpha\beta}$. If the opinion is represented by the parameters r and s, we use the notation $o = (r, s)^{rs}$.

The mode $t = mode(\alpha, \beta)$ of the distribution $Beta(\alpha, \beta)$, is given as:

$$t = mode(\alpha, \beta) = \frac{\alpha - 1}{\alpha + \beta - 2} = \frac{r}{r + s} \tag{2}$$

For any $c \in \mathbb{R} \setminus 0$ holds

$$mode((r, s)^{rs}) = mode((c \cdot r, c \cdot s)^{rs}) \ . \tag{3}$$

This representation allows for easily integrating feedback in the trust model. Assuming that feedback fb can be expressed as real number in $[-1; 1]$, where '-1' expresses a negative experience and '1' a positive one, the update of an opinion, can be done by recalculating the parameters $r_{new} = r_{old} + 0.5 * (1 + fb)$ and $s_{new} = s_{old} + 0.5 * (1 - fb)$ (cf. [6]). This representation enables to update the trust model without user interaction, if the feedback can be generated automatically. Furthermore, the software agents can use all statistical information, which can be derived from the beta distribution, such as mean value, variance, as basis for decision making.

3.3 Mapping Between Both Representations

Trust value t of an opinion $o = (\alpha, \beta)^{\alpha\beta}$ is defined as the mode of the corresponding beta distribution. The certainty value c of an opinion $o = (\alpha, \beta)^{\alpha\beta}$ in a context con_i is defined as follows: The maximal number of expected evidence can be denoted by $e(con_i) = \alpha_{max} + \beta_{max} - 2$, where α_{max} and β_{max} fulfill:

$$mean_{coll} := \frac{\alpha}{\alpha + \beta} = \frac{\alpha_{max}}{\alpha_{max} + \beta_{max}} =: mean_{max} \tag{4}$$

Then the certainty c is calculated as:

$$c = \frac{f(mean_{coll} \mid \alpha, \beta) - 1}{f(mean_{max} \mid \alpha_{max}, \beta_{max}) - 1} \tag{5}$$

Definition 3.1 (Mapping).
It holds $(\alpha, \beta)^{\alpha\beta} = (t, c)^{HTI}$, iff $t = mode(\alpha, \beta)$ and the certainty c fulfills Eq. 5.

Justification The mapping provides the translation between both representations. Therefore, the interpretation of an opinion in the HTI by users has to be as close as possible to the interpretation of the same opinion in the evidence model by a software agent, and vice versa. This way, it is possible that the user is able to interpret and adjust opinions, which are based on the feedback collected by the software agent, correctly.

Assuming, that a user has to set up the parameters for the trust value and certainty based on a countable number of experiments, then it seems to be intuitive that the user sets the trust value, which estimates the probability of for the expected event, close to the observed relative frequency. Since the mode of a pdf is equal to the relative frequency of the observed event, the trust value t is close to the intuitive interpretation of the user.

Similar to [5,10,12], the certainty value is intuitively linked to the number of collected evidence. That is, a greater number of collected evidence leads to higher confidence in the trust value, and therefore, to a higher certainty value. The maximal number of expected evidence $e(con_i)$, as introduced in Section 2.2, corresponds to the maximal certainty value. Similar to [12], we want the certainty to increase adaptively with the collected number of evidence. Therefore, we enforce that the first pieces of evidence increase the certainty value more, than later ones. As shown in Fig. 3 the certainty value as defined in 3.1 fulfills these properties. In absence of information ($r + s = 0$), the certainty value is $c = 0$, and $c = 1$ if the collected number of evidence is equal to the expected number of evidence. Between the two extremes, the certainty value increases adaptively. If the number of collected evidence is greater than the number of expected evidence, there is a normalization, which preserves the trust value and scales the certainty to $c = 1$ (see Eq. 6).

Furthermore, we can see from Fig. 3, that there is a slight dependency between the trust value t and the certainty value c. Since the final user interface is not continuous, but based on a small set of discrete values, this dependency can be neglected. Therefore, we consider the trust value and the certainty to be independent in the HTI, as demanded in 3.1.

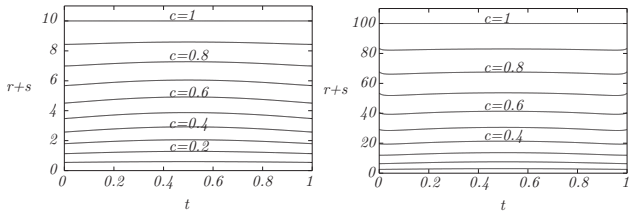

Figure 3: Iso-certainty lines for different maximal numbers of expected evidence: $e = 10$ (left), $e = 100$ (right)

We now provide a *mathematical interpretation* of the certainty parameter (see Fig. 4). Let p denote the probability of the experiment. The area over an arbitrary interval under the curve of the pdf can be interpreted as the probability that p is in this interval. Consider an interval with width w centered on the mean value of a pdf. The area in this interval represents the probability of the mean value. For opinions with the same mean value, this probability increases with an increasing amount of evidence. The certainty is the result of the comparison of this specific area based on the maximal number of expected evidence, to the same area based on the number of collected evidence. If we approximate those to areas by rectangles, with width w and height $h = f(\alpha/(\alpha + \beta) \mid \alpha, \beta) - 1$, the area can be denoted as $A = w \cdot h$. If we set A_{max} as the area which corresponds to the maximal number of expected evidence, and A_{coll} as the area which corresponds to the collected number of evidence, then $c = A_{coll}/A_{max} = (w \cdot h_{coll})/(w \cdot h_{max}) = h_{coll}/h_{max} = (f(mean_{coll} \mid \alpha_{coll}, \beta_{coll}) - 1)/(f(mean_{max} \mid \alpha_{max}, \beta_{max}) - 1)$. To justify the approximation of the area with a rectangle consider w to be small enough.

Normalization If the opinion $o = (r, s)^{rs}$ of an agent is based on a greater number of evidence than the maximal number of expected evidence, the collected number of evidence will be scaled to the allowed maximum (see Eq. 6). The normalization preserves the mode of the pdf (see

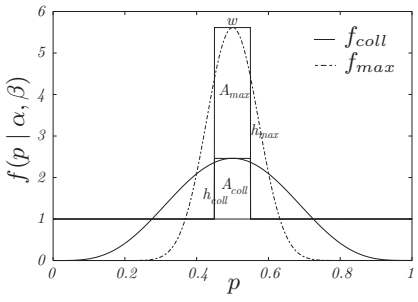

Figure 4: Visualization for the mathematical interpretation

Eq. 3), and therefore, does not change the trust value. The normalized opinion $norm(o)$ will be used as input for the discounting described above.

$$norm((r,s)^{rs}) = \begin{cases} (r,s)^{rs} & \text{if } r+s \leq e \ , \\ (\frac{r}{r+s} \cdot e, \frac{s}{r+s} \cdot e)^{rs} & \text{else } . \end{cases} \quad (6)$$

3.4 Trust Propagation in the Evidence Model

For trust propagation we define two operators, similar to the ones defined by Jøsang in [5]. We also call our operators 'consensus' for the aggregation of opinions and 'discounting' for the recommendation of an opinion. The consensus operator is identical with the one presented in [5]. Since the discounting operator presented by Jøsang is motivated based on the belief model, we decided to define a new operator, which is motivated based on the evidence model.

DEFINITION 3.2 (CONSENSUS). *Let $o_x^A = (r_x^A, s_x^A)^{rs}$ and $o_x^B = (r_x^B, s_x^B)^{rs}$ be the opinions of A and B about the truth of the proposition x. The opinion $o_x^{A,B} = (r_x^{A,B}, s_x^{A,B})^{rs}$ is modeled as the opinion of an imaginary agent which made the experiences of A and B, and is defined as:*

$$o_x^{A,B} = o_x^A \oplus o_x^B = (r_x^A + r_x^B, s_x^A + s_x^B)^{rs} \quad (7)$$

The $'\oplus'$ symbol denotes the consensus operator. The operator can easily be extended for the consensus between multiple opinions.

Justification The result of the consensus is the opinion of an agent, who has made the observations done by A and the observations done by B.

DEFINITION 3.3 (DISCOUNTING). *Let $o_B^A = (r_B^A, s_B^A)^{rs}$ and $o_x^B = (r_x^B, s_x^B)^{rs}$. We denote the opinion of A about x based on the recommendation of B as o_x^{AB} and define it as:*

$$o_x^{AB} = o_B^A \otimes o_x^B = (d_B^A r_x^B, d_B^A s_x^B)^{rs} \ , \text{where } d_B^A = t_B^A c_B^A \ . \quad (8)$$

The $'\otimes'$ symbol denotes the discounting operator. In a chain of recommendations, we start at the end of the chain, e.g., $o_x^{ABC} = o_B^A \otimes (o_C^B \otimes o_x^C)$.

Justification The discounting reduces the number of evidence taken into account, since $d_B^A \in [0;1]$. The discounting factor d_B^A increases with a number of positive evidence. That is, if A and C have the same amount of total evidence with B, but A has more positive evidence, then A gives a stronger weight (i.e., a larger discount) to the recommendation of B than C.

Furthermore, the discounting factor increases with the number of collected evidence. That is, if A and C have the same ratio of positive and negative evidence with B, but A has more evidence in total, then A has more evidence to believe that B will behave as in the past. Therefore, A gives the opinion of B a stronger weight than C does.

3.5 Trust Propagation in HTI

For the propagation of trust, we transfer the representation of a recommendation to the evidence-based representation and use the operator for consensus and discounting as defined above. The consensus operator increases the certainty of the resulting opinion, if multiple recommendations are provided. The discounting operator only decreases the certainty of a recommendation, but preserves the trust value (mode) of a recommendation. Assume that $o_x^{AB} = o_B^A \otimes o_x^B$, then it holds $c_x^{AB} \leq c_x^B$ and $t_x^{AB} = t_x^B$.

4. RELATED WORK

Trust is addressed by a growing group of researchers. Focusing on trust modeling, there is a multitude of different approaches, considering the representational models of trust, and the algorithms for handling recommendations [11].

The seminal work in the field has been done by Marsh [9]. His trust model is based on the social aspects of trust. It includes importance and utility of a situation in the computational model. The decision making is threshold based and considers trust as well as risk. The main drawbacks of this model are that it models trust one-dimensional, and that it focuses on trust based on direct experience, but does not deal with recommendations.

There are several other approaches which model trust one-dimensional, e.g., TidalTrust [4] and EigenTrust [8]. In those models, trust is represented by a single trust value, which does not allow to express the certainty or the reliability of this trust value. Therefore, those models cannot express whether an opinion is based on a single piece or multiple pieces of evidence. This leads to a loss of information, when recommendations are aggregated to a single value. Furthermore, problems may arise when interpreting recommendations. For example, if, in TidalTrust, an agent receives only recommendations from lowly trusted recommenders, the aggregated trust value does not reflect, that it is based on lowly trusted recommenders.

If the aggregated trust value would be decreased, to express the low trust in the recommender, it would be hardly possible to distinguish this opinion from one, which is provided by a highly trusted recommender who has recommended a low trust value.

Other approaches model trust with two or three dimensions. Two dimensional trust models are often based on the Bayesian approach, e.g., [2, 3, 5]. Those models do not have an explicit parameter for certainty or uncertainty. It has to be derived from the beta probability density function. As stated in [7], those models are often too complicated to be well-understood by average users.

The trust models presented in [2, 3, 5] also allow for representations as belief model. The belief model approaches use the triple belief b, disbelief d, and uncertainty u to represent trust. The drawback of belief models is that the three parameters cannot be assigned independently, e.g., in [5]

they are interrelated by $b + d + u = 1$. Thus, the presence of uncertainty influences both belief and disbelief. It is non-trivial for users to express, e.g., a medium belief with different levels of uncertainty. In our model, it is possible to independently choose the values for trust and certainty. The relation between the belief b and disbelief d as defined in [5] and our trust value can be denoted as $t = b/(b + d)$.

Approaches like the 'Subjective Logic' [5] are not capable of expressing uncertainty context-dependent. In [5], the uncertainty u is defined as $u = 2/(r + s + 2)$. Therefore, uncertainty depends only on the number of collected evidence, but not on the context. If we choose the maximal number of expected evidence as $e = 10$ or $e = 20$, then the uncertainty in 'Subjective Logic' behaves similar to a parameter $(1 - c)$, which can be used to express uncertainty in our model. If we think of contexts related to a high risk or a high frequency of interaction, and assign, e.g., the maximal number of expected evidence $e = 100$ to this context, the uncertainty in 'Subjective Logic' does not any longer behave as expected, since opinions based on 18 or more collected evidence ($r + s \geq 18$) have only very little uncertainty ($u \leq 0.1$).

The trust models presented in [10,12] introduce *reliability* as a concept which is similar to our concept of *certainty*. They also define a context-dependent value similar to the maximal number of expected evidence.

In [10], the trust model is based on the Bayesian approach, as ours. The maximal number of expected evidence e corresponds to m, which is described as the "minimal number of encounters necessary to achieve a given level of confidence [...]". The reliability w is defined to increase linear with the number of collected evidence from 0 (if no evidence is available) to 1 (if the number of collected evidence is greater than or equal to m). But, this linear approach is stated to be an first order approximation [10].

In the trust model presented in [12], the *intimate* level of interactions, is close to the concept of the maximal number of expected evidence. It is also content-dependent. The *number of outcomes factor* (No) $\in [0, 1]$, increases adaptively with the number of collected evidence. To achieve the adaptive behavior as described in 3.3, Sabater et al. put the ratio between the collected and the expected number of evidence in a *sinus*-function, which seems to be done ad hoc.

5. CONCLUSION & FUTURE WORK

In this paper we developed a trust model, which allows to represent trust in a way, which can be interpreted and updated by software agents as well as by users. We provided a new way to express the certainty of an opinion in contexts, which are associated with different levels of risk, or frequency in interaction. In the HTI the values for trust and certainty can be interpreted independently, which allows to introduce the semantics of an opinion based on labels. Therefore, the user is able to easily control the state of the trust model and to adjust opinions, if necessary. The evidence model enables software agents to update an opinion, when new evidence is available, and to reason about the trustworthiness of an interaction partner. Furthermore, we showed that our mapping between the HTI and the evidence model has an intuitive interpretation and is mathematically founded. We provided two operators for trust propagation, which are motivated based on the evidence model and have an intuitive interpretation in the HTI representation.

Our future work will include the development of trust management and decision making strategies. Those are necessary to be able to evaluate the trust model in a simulation, and to enhance the attack-resistance of the model. Furthermore, we will refine the discrete representation for the human trust interface based on user studies.

6. REFERENCES

[1] B. Bhargava, L. Lilien, A. Rosenthal, and M. Winslet. Pervasive trust. *IEEE Intelligent Systems*, 19(5):74–88, 2004.

[2] V. Cahill et al. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, 2/3:52–61, July 2003.

[3] M. Carbone, M. Nielsen, and V. Sassone. A formal model for trust in dynamic networks. In *Proc. of IEEE International Conference on Software Engineering and Formal Methods*, September 2003. IEEE Computer Society.

[4] J. Golbeck. *Computing and Applying Trust in Web-Based Social Networks*. PhD thesis, University of Maryland, College Park, 2005.

[5] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.

[6] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, June 2002.

[7] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. In *Decision Support Systems*, 2005.

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. of the 12th international conference on World Wide Web*, pages 640–651, New York, USA, 2003. ACM Press.

[9] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

[10] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7*, page 188, Washington, DC, USA, 2002. IEEE Computer Society.

[11] S. Ries, J. Kangasharju, and M. Mühlhäuser. A classification of trust systems. In *Proceedings of the International Workshop on MObile and NEtworking Technologies for social applications*, 2006.

[12] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 475–482, New York, NY, USA, 2002. ACM Press.

[13] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, September 2005.

[14] J.-M. Seigneur, S. Farrell, C. D. Jensen, E. Gray, and Y. Chen. End-to-end trust starts with recognition. In *First International Conference on Security in Pervasive Computing*, Mar. 2003.