

TRUST MODEL FOR HIGH QUALITY RECOMMENDATION

G. Lenzini, N. Sahli and H. Eertink

Telematica Instituut, Brouwerijstraat 1, 7523 XC, Enschede, The Netherlands

Keywords: Trust Model, Recommendation, Argumentation, Agent.

Abstract: In this paper, we propose a trust management model for decentralised systems that improves the *quality of recommendations* that members of a virtual community get about the trustworthiness of objects. In our system, as in well-known solutions, members of the community evaluate (i) the functional trust in an item by the analysis of the object's qualities, past experience, and recommendations and (ii) the referral trust in a recommender by the analysis of the recommender's qualities and reputation based on personal experience. Moreover, in our trust model, each principal debates with its recommenders about the justifications given to support a recommendation. Thus, the usefulness and the reliability of a recommendation depend also on the strength of the arguments supporting the recommendation. A measure of this strength results after the member has played an argumentation game with the recommender. Therefore, the recommendations that are taken into account are those which better match the member's profile and way of reasoning. Our trustworthiness evaluation algorithm is context dependent and able to collect both direct and indirect information about trustees. Our trust model is part of an agent-based architecture we propose for decentralised virtual communities. This architecture provides our system with autonomy, unobtrusiveness, user mobility, and context-awareness.

1 INTRODUCTION

A virtual community (also e-community or on-line community) is a group of socially interacting people whose interaction is supported by computer systems (Preece, 2000). People interact primarily with e-mails, chats, etc. rather than face to face. One common application of virtual communities is sharing opinions about objects of interest, for example recommendations and ratings. By using opinions, each member can take a shortcut to items she/he likes without having to try them or to experience many similar items. Unsurprisingly, this facility has become popular on the Internet. "Amazon", for example, rates each item with stars. This overall rating is obtained by averaging the ratings provided by users on the quality of the product as a whole. Any member can write a review. In IENS.nl, a Dutch site for rating restaurants, and apart from the overall rating of a given restaurant, a user can also rate/consult a refined set of criteria related to the restaurant such as quality of food, quality of service, decor, and price. Such systems (i.e., Amazon, IENS, e-Bay) are mainly centralised; they rely on all peers' ratings and give the same recommendations

to all peers. In these systems, the inference process that leads to a recommendation (either in the form of a rating or a suggestion) is usually hidden to the user. It is not possible to perform a *qualitative* analysis on the reliability of the suggestion provided nor on the process that has been used to compose it. It is also not possible to personalise the ratings w.r.t. the user's way of reasoning. As a consequence, a user who is looking for a recommendation which fits his/her taste must read a large number of reviews left by the other users. In fact, she/he has to manually look for justifications that support the choice which better matches his/her profile and expectations.

A decentralised approach has thus emerged as a serious alternative (e.g., see (Sabater and Sierra, 2002; Miller et al., 2004; Teacy et al., 2005)). When the trustworthiness and the personalisation of recommendations is an issue of great concern, decentralisation might be more appropriate. In a decentralised approach the user only relies on recommendations suggested by other peers in whom she/he trusts. We propose an agent-based decentralised system for trust-based recommendations which goes farther than previous decentralised systems. Our system aims at im-

proving the *quality of recommendations*. Since even these recommenders could have different tastes and different ways of judging, being trustworthy is not enough to have suggestions accepted by the current peer. Thus, we propose that the peer should also debate with its recommenders about the arguments that justify a recommendation in order to only select the most likely appropriate (from this peer's view point) recommendations.

In this paper we focus on the argumentation process and its impact on the trustworthiness of recommendations. The remainder of the paper is organised as follows. Section 2 gives an overview of the main principles of our approach that are behind the quality of recommendations. Section 3 explains the rules that model our trust evaluation system with focus on the argumentation process. Section 4 explains how to measure the "strength" of a recommendation. Section 5 illustrates an example of the use of trust evaluation with argumentation. Section 6 presents the agent-based architecture we propose to support our trust model. Section 7 comments the related work and Section 8 concludes the paper and points out the future work.

2 OUR APPROACH

In (Lenzini et al., 2008) we presented a preliminary solution to improve the quality of recommendations in mobile and non-mobile open virtual communities. A community is hosted in what we call a *Virtual Agora* (in short, VA). A VA is a virtual open space (e.g., web site, server) where active entities (e.g., consumers) meet, interact, and share experiences about items (e.g., goods, services) of interest. From this point of view, many services already available match the definition of a VA (e.g., the mentioned Amazon and e-Bay). Moreover, a VA satisfies also the following characteristics: (i) *openness*, entities from various sources can freely join or leave at any time; (ii) *decentralisation*, no central authority controls entities (but we admit a certain centralised facility to search for members, see Section 6), and (iii) *persistence*, entities (if desired) can be continuously available. In this paper, we present an advanced solution for the improvement of the quality of recommendations within a VA. It is based on the following three ideas:

- (1) In the VA, each user is represented by a software agent, called *delegate agent*, which behaves on behalf of its user (even when the user is off-line)
- (2) Each *delegate agent* maintains a personal register of rated items (*Register of Rated Items*) and a

personal trust-weighted register of recommenders (*Register of (Un)Trusted Recommenders*).

- (3) When evaluating the trustworthiness of an item, a *delegate agent* accepts only high quality recommendations. The quality of a recommendation depends upon both the referral trust of the source (from this agent's point of view) and the measure of the "strength" of the arguments that support the recommendation.

We now comment the three aforementioned ideas.

Point (1) concerns the architectural design of a VA, which is part but not the main goal of this paper. Briefly, peers need autonomy to be able to act on behalf of their users. To design an effective VA which fulfils this requirement, we chose an agent-based architecture. Each peer is represented by a *delegate agent* that once moved to the VA is constantly interacting with the other *delegate agents* and updating its user's registers of trust. More details of the architecture are given in Section 6.

Point (2) concerns the two trust-weighted registers that each *delegate agent* uses to evaluate the functional trust on an item and the referral trust on a recommender. In the register called *Register of Rated Items* (in short *Trat*), an agent keeps trace of the functional trust in the items that it has evaluated so far, directly (from personal experience), indirectly (reported by other recommenders) or both. In the register called *Register of (Un)Trusted Recommenders* (in short *Trec*) an agent keeps trace of its referral trust on other peers in giving recommendations. The two registers are initiated and updated according to rules that are explained in Section 3.

Finally, point (3) regards the quality of recommendation. When a peer (i.e., *delegate agent*) asks for recommendations it evaluates the information it receives according to the referral trust of the source and to the "appropriateness" of the recommendations with respect to its profile, taste, and way or reasoning. The *delegate agent* reaches this latter goal by "discussing" with its recommenders about the arguments that they provide to justify their recommendations. Eventually, whether or not the agent accepts a recommendation, depends upon both the warrant of the recommendation and the referral trust of the recommender. Thus, "warranted" recommendations coming from not so trustworthy members may be accepted, whereas "unwarranted" recommendations coming from trustworthy recommenders might be discarded. We believe that an opinion built upon a small number of recommendations of high quality is generally more useful than a digest of anonymous, often contradicting, and potentially unjustified recommendations.

3 TRUST RELATIONSHIPS MANAGEMENT

This section explains our trust model and the algorithm that a peer uses to build and to update its *Trat* and *Trec*. The basic rules have already been presented in (Lenzini et al., 2008). Here, we introduce and study new rules whose goal is to quantify the quality of a recommendation in the trust evaluation process. We also present an implementation based on *argumentation* theory (e.g., see (Prakken, 2006)). We introduce and discuss the concept of *argumentation trust*, which we use as a measure of the recommendation strength. To our best knowledge, the framework we present is innovative and definitely more dynamic than our previous solution.

The attention to the justification of a recommendation comes from the need of distinguishing between a measure of the reputation of a recommender (in giving honest recommendations) and the appropriateness of a specific recommendation in a given context. For example, even if peer *B* is renowned to be an expert in restaurants (fact that she/he has proved many times), her/his recommendation about going to the Italian restaurant “La Barcarola” may be discarded by a principal *A* if *A* and *B* have different opinions about what a good Italian menu should be. What is excellent from *B*’s view point could be inappropriate for *A* (which has, in this particular context, different preferences). On the other hand, *A* may accept the recommendation of another peer *D* which is less qualified than *B* but whose justification about going to the “L’Ostricarò”, another Italian restaurant, better convinces *A*. Later, when *A* effectively tries this restaurant and likes it, *D* also sees its trustworthiness growing from *A*’s point of view; *D*’s recommendation has perfectly satisfied *A*’s expectations. It is worth noticing that *B*’s reputation remains intact. *A* has no interest in saying that *B* is a bad recommender only because *A* has not followed *B*’s advice. It is not *B*’s reputation that is in question here, but rather the justification that *B* gave about her/his recommendation, which didn’t match *A*’s way of reasoning from the beginning.

The *Register of Rated Items* of a member *A* (written *Trat(A)*) is a set of trust relationships, written as

$$A \xrightarrow[\mathcal{G};\mathcal{C}]{(i,m)} \sigma b$$

Here, *A* is an agent and *b* is an item. All *b*’s that are in *Trat(A)* have been evaluated by *A* directly, indirectly, or both. The value *m* is a measure of the functional trust (positive or negative) that *A* has in *b*. The natural number *i* is a time-stamp from which to deduce the age of the trust relation. The subscript *G*

is the set of recommendations and of recommenders that have been consulted in composing *m*. The subscript *C* is the context in which the trust relation takes or has taken place. This parameter makes the trust relationship context-dependent. If *b* is a restaurant, for example, one’s evaluation may change depending on the social context of the user (e.g., with colleagues, with partner, or with family). The subscript σ stands for a trust *aspect* (called trust scope in (Jøsang et al., 2006)). It represents the purpose of the trust relation. In *Trat(A)* there is a multiplicity of trust relationships between *A* and *b* for each aspect among σ_1, σ_2 , etc. To simplify our exposition, we consider only one σ which we omit from our notation. Anyhow, it must be clear that the overall trust relationship between *A* and *b* depends upon all the trust *aspects*.

The *Register of (Un)Trusted Recommenders* of a member *A* (called *Trec(A)*) contains trust relationships written as

$$A \xrightarrow[\mathcal{O}]{(i,\omega)} D$$

Here *A* and *D* are both agents. If a member *D* is in *Trec(A)*, this means that *D*’s trustworthiness has been evaluated by *A*. The natural number *i* is a time-stamp, and the value ω represents an amount of referral trust that *A* has in *D*. Set *O* carries information about the items that *D* has so far recommended to *A* and the relative recommendations.

3.1 Trust Evaluation Algorithm

The process of trust evaluation is formalised with an inference system, which is part of an agent *A*’s intelligence. Figure 1 and 2 summarise the rules of the inference system used to build *Trat(A)* and *Trec(A)*, respectively. Each trust evaluation rule, in *A*’s intelligence, has form: $\frac{\text{premises}}{\text{conclusion}}$ *A conditions* *Premises* are required in order to infer the *conclusion*. The inference mechanism is processed if *conditions* (if present) are fulfilled. The subscript, here a generic agent *A*, indicates that this rule is applied by *A*. We will omit the subscript in the rest of the paper since we present all rules from *A*’s point of view.

As a matter of computation, we use two categories of arrows, namely *temporary* (or tentative), written $\bullet \rightarrow$ and $\bullet \Rightarrow$ resp., and *eventual* (or conclusive), written \rightarrow and \Rightarrow respectively (see Figures 1 and 2). A *temporary* trust relation emerges from an incomplete trust evaluation; additional inputs can still affect the trust value. An *eventual* trust relation indicates that the trust evaluation has ended; no more input can change the trust value for time *i*. The registers *Trat(A)* and *Trec(A)* are composed of only *eventual* trust relationships. In the following, we assume an algebra

$$\begin{array}{c}
 \text{Bootstrapping} \\
 (1) \\
 \begin{array}{ccc}
 (a) \frac{[\text{eval}_A(\{b_i\}, C) = m]}{A \xrightarrow{(i,m)} b} & (c) \frac{[\text{tag}_A^i(b, C) = m]}{A \xrightarrow{(i,m)} b} & (b) \frac{[\text{eval}_A(\{b'_i: b' \sim b\}, C) = m]}{A \xrightarrow{(i,m)} b} \\
 \{(A_i, m)\}; C & \{(A, m)\}; C & \{(A_i, m)\}; C
 \end{array} \\
 \boxed{\text{Recommendation}} \\
 (2) \\
 \begin{array}{ccc}
 (a) \frac{A \xrightarrow{(i,\omega)} D \quad D \xrightarrow{(i-1,m)} b \quad A \xrightarrow{(i,\omega')} D}{A \xrightarrow{(i,\omega' \otimes m)} b} & \omega \oplus \omega' = \omega'' > \omega_0 & (b) \frac{A \xrightarrow{(i,\omega)} D \quad D \xrightarrow{(i-1,m)} b' \quad A \xrightarrow{(i,\omega')} D}{A \xrightarrow{(i,\omega' \otimes (\omega_{d(b',b)}) \otimes m)} b} \\
 & & b' \sim b, b' \neq b \\
 & & i > 0 \\
 & & \omega \oplus \omega' = \omega'' > \omega_0 \\
 (c) \frac{A \xrightarrow{(i,\omega)} D \quad D \xrightarrow{(i-1,m)} b \quad A \xrightarrow{(i,\omega')} D}{A \xrightarrow{(i,\omega' \otimes (\omega_{d(C',C)} \otimes m))} b} & C' \sim C, C' \neq C \\
 & i > 0 \\
 & \omega \oplus \omega' = \omega'' > \omega_0 & (d) \frac{A \xrightarrow{(i,\omega)} D \quad D \xrightarrow{(i-1,m)} b \quad A \xrightarrow{(i,\omega')} D}{A \xrightarrow{(i,\omega' \otimes m)} b} \\
 & & G \neq \{(D, m)\} \\
 & & A \notin G_{i1} \\
 & & i > 0 \\
 & & \omega \oplus \omega' = \omega'' > \omega_0
 \end{array} \\
 \text{Experience} \\
 (3) \\
 \begin{array}{ccc}
 (a) \frac{A \xrightarrow{(k,m)} b}{\{(A, m)\}; C} & \omega = \omega_{i-k} \\
 & k < i & (b) \frac{A \xrightarrow{(k,m)} b'}{\{(A, m)\}; C} \\
 & & b' \sim b \\
 & & b \neq b' \\
 & & \omega' = \omega_{d(b',b)} \\
 & & \omega = \omega_{i-k} \\
 & & k < i \\
 (c) \frac{A \xrightarrow{(k,m)} b}{\{(A, m)\}; C'} & C' \sim C \\
 & C \neq C' \\
 & k < i
 \end{array} \\
 \text{Finalising} \\
 (4) \\
 \begin{array}{cc}
 (a) \frac{A \xrightarrow{(i,m)} b \quad A \xrightarrow{(i,m')} b}{A \xrightarrow{(i,m \oplus m')} b} & G_{i1} \cap G'_{i1} = \emptyset \\
 (b) \frac{A \xrightarrow{(i,m)} b}{A \xrightarrow{(i,m)} b} & G; C
 \end{array}
 \end{array}$$

Figure 1: Inference rules for the management of agent-item trust relationships. The recommendation rules, whose title is boxed, and their implementation (in terms of argumentation) form the original contribution of this paper.

($TValues, \oplus, \otimes, 0, 1$) of trust values. The binary operator \oplus is used to merge trust values, whilst \otimes to discount trust. Values 0 and 1 are the respective neutral elements. The literature offers many computational models that are in principle applicable.

Building and Updating *Trat*. Referring to Figure 1, rules (1.a)-(1.b) allow A to apply its own procedure $\text{eval}_A(x, y)$ (whose concrete implementation is left unspecified at this level of abstraction) to estimate b 's trustworthiness. The procedure takes as input the following two parameters: b_i , which with a little abuse of notation represents the information available about b at time i and in context C . In case of information about b , a witness b' (i.e., $b' \sim b$) can be used instead of b . The label $\{(A_i, m)\}; C$ in the resulting trust relationship records that the value m has been

calculated at time i and in context C . Rule (1.c) models the action of rating b which follows a direct experience. A uses the function $\text{tag}_A^i(x, y)$ to tag b at time i and in context C . The trust relation that emerges after a tagging is an *eventual* relation; there is no need of other information to evaluate b 's trustworthiness.

Rules (2.a)-(2.d) in Figure 1 describe how we manage recommendations. These rules extend and renew the corresponding rules we gave in (Lenzini et al., 2008), where we allowed an agent A to accept a recommendation when the recommender's referral trust is above a certain threshold and when the justification given for the recommendation is accepted by A . We had there no quantitative evaluation of the justification of a recommendation. In Figure 1, instead, the trustworthiness of D 's recommendation is evaluated

$$\begin{array}{c}
 \text{Bootstrapping} \\
 (a) \frac{[\text{estimate}_A(D_i) = \omega]}{A \xrightarrow[\emptyset]{(i,\omega)} D} \quad (b) \frac{A \xrightarrow[\emptyset]{(i-1,\omega)} D}{A \xrightarrow[\emptyset]{(i,\omega)} D} \quad i > 0 \\
 \\
 \text{Update} \\
 (a) \frac{A \xrightarrow[\emptyset]{(i,\omega)} D \quad A \xrightarrow[\{(A_i,m')\};C]{(i,m')} b \quad A \xrightarrow[\mathcal{G};C]{(k,m'')} b}{A \xrightarrow[\emptyset \cup \{(b,k)\}]{(i,\text{inc}(\omega))} D} \quad \begin{array}{l} k \leq i, \\ (b,k) \notin \mathcal{O} \\ (\{(D,m)\} \in \mathcal{G}) \wedge \\ (m' \sim m) \end{array} \quad (b) \frac{A \xrightarrow[\emptyset]{(i,\omega)} D \quad A \xrightarrow[\{(A_i,m')\};C]{(i,m')} b \quad A \xrightarrow[\mathcal{G};C]{(k,m'')} b}{A \xrightarrow[\emptyset \cup \{(b,k)\}]{(i,\text{dec}(\omega))} D} \quad \begin{array}{l} k \leq i, \\ (b,k) \notin \mathcal{O} \\ (\{(D,m)\} \in \mathcal{G}) \wedge \\ (m' \not\sim m) \end{array} \\
 \\
 \text{Finalisation} \\
 (a) \frac{A \xrightarrow[\emptyset]{(i,\omega)} D \quad A \xrightarrow[\emptyset']{(i,\omega')} D}{A \xrightarrow[\emptyset \cup \emptyset']{(i,\omega \oplus \omega')} D} \quad \emptyset \cap \emptyset' = \emptyset \quad (b) \frac{A \xrightarrow[\emptyset]{(i,\omega)} D}{A \xrightarrow[\emptyset]{(i,\omega)} D}
 \end{array}
 \tag{7}$$

Figure 2: Inference rules for managing agent-agent trust relationships.

by taking into account both the referral trust (that A has already in D) and a measure of trust that emerges from A 's analysis of the arguments that D gives to justify its recommendation. The relation $A \xrightarrow[\emptyset]{(i,\omega)} D$ indicates that A 's evaluation of D 's recommendation (after having discussed about its justification) is ω . We call this value *argumentation trust*. Section 4 explains how we estimate argumentation trust. Function $\omega \oplus \omega'$ reflects a merged value between ω (argumentation trust) and ω' (referral trust) in our algebra of trust values. As a matter of example, it may happen that A accepts a recommendation from a peer that is not so highly trustworthy (e.g., a new member) but whose justification is strong. Similarly, a recommendation that comes from a highly trustworthy recommender but whose justification is unconvincing, can be refused. The resulting mechanism for (referral) trust evaluation is more flexible. The “strength” of justifications (given by a recommender) is thus as important as the reputation of the recommender.

Rules (3.a)-(3.c) show how to use the experience of A about b , or about some of b 's witnesses (i.e., $b' \sim b$), to evaluate b 's temporary trust. Information related to the same context, or to a compatible context (i.e., $C' \sim C$), can also be considered. When witnesses or compatible contexts are used, the past experience is discounted depending on a measure of distance between b' and b or between C' and C , respectively (see also (Toivonen et al., 2006)).

Rule (4.a) describes how the temporary evaluations collected by different sources can be merged.

The condition $\mathcal{G}_{i1} \cap \mathcal{G}'_{i1} = \emptyset$ informally says that the sources must be disjoint to avoid interferences. Finally, rule (4.b) finalises the trust process. The *temporary* trust evaluation becomes *eventual*. The recommenders, as well as their recommendations that have been used in estimating b 's trustworthiness, are logged in \mathcal{G} . They are used to update $Trec(A)$ (see next paragraphs).

Building and Updating $Trec$. Referring to Figure 2, rule (5.a) initiates $Trec(A)$. D 's profile at time i is evaluated by A with the function $\text{estimate}_A(x)$. The value ω is an estimation of D 's referral trustworthiness. Implementations of $\text{estimate}_A(x)$, out of the scope of this paper, can be based on the amount of “likelihood” between D and A for example, or on the number of items that have been similarly ranked by A and D . Rule (5.b) expresses that trust can be estimated by inheritance with a previous trust. Here we do not model *trust decay* over time, or in other words we do not discount a referral trust because of time.

Rule (6.a) and rule (6.b) say that A 's (referral) trust in an agent D is increased (resp., decreased) for each *compatible* (resp. *incompatible*) recommendation given by D . A recommendation is compatible only in comparison with A 's personal experience (i.e., tagging). Here, overriding our notation, the symbol \sim indicates the compatibility relation among opinions. After A tags b , its trust in b is more based on facts than on presumptions (according to our notation, this is recognisable from the subscript $\{(A_i, m)\}$). Therefore, A is able to review the opinion she/he got via rec-

ommendation and to modify her/his trust in the recommender accordingly. The comparison is context-dependent. Rule (7.a) says that the temporary evaluations collected when updating the referral trust for a recommender can be merged (operator \oplus) together. A recommender might have given opinions on more than one item and its overall referral trust depends on the trust update for all of them. The condition $O \cap O' = \emptyset$ ensures that the same update (per item) is not considered twice. Finally, rule (7.b) finalises the trust process. The *temporary* trust evaluation becomes *eventual*, and it will become part of $Trat(A)$. The items recommended along the way that contributed to the referral trust are logged.

Implementation of the Model. To apply the rules for $Trat(A)$ and $Trec(A)$ in an effective trust evaluation algorithm, an order of preference among rules is needed. Different ordering constraints reflect different attitudes in merging opinions and past experiences. For example, an agent may prefer to first collect all opinions, then to merge them, and finally to consider her/his past experience. In rules (3.a)-(3.c) for example, different constraints over the time variable k guide the search strategy in the past (e.g., choosing a maximal k implies considering the most recent experience). Discussing these choices is out of the focus of this paper.

4 ARGUMENTATION TRUST

This section explains how we measure the strength of a recommendation given a set of arguments. We first remind the basic concepts of argumentation. In logic and in artificial intelligence, argumentation is a way of formalising common-sense reasoning. Current implementations use both classical, monotonic, logic and non-monotonic reasoning. Depending on the implementation, an argument can be presented as an inference tree, a deduction sequence, or a pair (*set of premises, conclusion*). Most of the argumentation systems can be characterised by five elements, namely, (i) an underlying *logical language*, (ii) a concept of *argument*, (iii) a concept of *conflict* between arguments (which involves counter-arguments that *attack* an original argument), (iv) a notion of *defeat* among arguments, and (v) a notion of *acceptability* of arguments according to a well-defined criterion.

A counter-argument can *attack* an original argument directly, its premises, or the rules that have been used to prove it. The right definition of *attack* depends on the chosen implementation. In (Bentahar and Meyer, 2007), which uses an implementation

based on classical logic, a counter-argument negates an original argument or one of its premises. In (Garcia and Simari, 2002), where a *Defeasible Logic* (Nute, 1994) implementation is used, counter-arguments are supported by stronger proofs or stronger knowledge than those supporting an original argument.

Attached arguments can be defended, for example by producing a counter-argument against the counter-argument. This interaction is expressed through and *argumentation protocol*, where attacks and defences interleave until the original argument is *accepted*, *refused*, or declared *undecided*. These three concepts will be used to quantify the strength of a recommendation (see Definition 1).

The integration we present is general and independent from the implementation chosen for the argumentation system.

Definition 1 (Argumentation Trust). *Let us suppose two agents, A and D, and an argumentation system Γ . A and D run N argumentation protocols to debate over the arguments that D provides to justify her/his recommendation. Let N_a , N_r , N_u ($N_a + N_r + N_u = N$) be the number of D's initial arguments that A accepts, refuses, and neither accepts nor refuses (i.e., undefined), respectively. Then the argumentation trust that A has on D is calculated with the following function:*

$$\omega(N) = \frac{N}{N_{au} + N_r + \left(\frac{1}{N_{au} + N_r + 1}\right)} \quad (8)$$

where $N_{au} = N_a + N_u$ is the number of arguments that A has either accepted or not refused.

In (8), high trust values are reached with a high number of *not refused* arguments. Thus, justifications made of a large number of arguments can bring to higher argumentation trust than justifications made of a small number of arguments (see Figure 3).

Referring back to Figure 1, $A \xrightarrow{i,\omega} D$ expresses that A and D have run an argumentation protocol and, as a result, A's *argumentation trust* in D is $\omega(N) = \omega$.

5 RUNNING EXAMPLE

In this section we illustrate an example of trust evaluation with argumentation. The argumentation game is informally described. This means that this example is not binded to a specific implementation. The instantiation of argumentation into a formal scheme is left as future work.

Let us assume that a member (Alice) of a slow food community is looking for information about

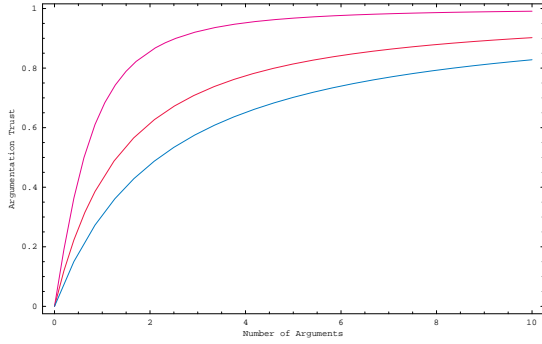


Figure 3: Graphic showing the argumentation trust function when all, all but one, all but two arguments provided by D are accepted by A , respectively (e.g., $N_r = 0, 1, 2$ from top to bottom). The more accepted arguments D provides the higher its argumentation trust can be.

restaurant “La Bella Vita” (in short, b) in Amsterdam. Alice wants to have a romantic dinner eating Italian food. Her *delegate agent* A , which runs in the Virtual Agora (i.e., the meeting infrastructure for the slow food community), will collect information about the trustworthiness of b . In the following, we assume that the algebra of trust values is the real interval $[0, 1]$ with a consensus operator $x \oplus y = (x + y)/2$ and a discount operator $x \otimes y = x \cdot y$. Trust values in $[0, 1]$ are informally read as follows: $[0, 0.3] =$ distrust, $(0.3, 0.5] =$ weak distrust, $(0.5, 0.7] =$ weak trust, and $(0.7, 1.0] =$ trust.

The goal of A is to evaluate the functional trust of b at the current time (let us say w_4). Formally, A has to infer $A \xrightarrow{(w_4, m)} b$, where C is the social context “romantic dinner”. By consulting its register of rated items $Trat(A)$, A finds out that: (i) it has never rated b before, but that (ii) it has evaluated restaurant “La Bella Vita” in Utrecht (in short, b') which belongs to the same franchising as b . This rating was registered when Alice was in Utrecht last week (week w_3) for a romantic dinner. The stored rating (functional trust in b') is 0.8. Formally, this means that $A \xrightarrow{(w_3, 0.8)} \{(A_{w_3}, 0.8)\}; C$ $b' \in Trat(A)$.

Because b and b' belong to the same franchising (i.e., $b' \sim b$), A estimates its trust in b by applying rule (3.b). If we assume that the trust decay is $\omega_{w_4-w_3} = 1.0$ (i.e., one week-old experience is fully significant) and the discount due to the use of a witness $\omega_{d(b', b)} = 0.94$ (i.e., the trust experience with a restaurant of the same chain is quite significant), then A 's temporary trust in b is $1.0 \cdot (0.94 \cdot 0.8) = 0.75$. Formally:

$$\frac{A \xrightarrow{(w_3, 0.8)} b' \quad \{(A_{w_3}, 0.8)\}; C}{A \bullet \xrightarrow{(w_4, 0.75)} b \quad \{(A_{w_3}, 0.75)\}; C} \quad \begin{array}{l} b' \sim b \\ \omega_{w_4-w_3} = 1.0 \\ \omega_{d(b', b)} = 0.94 \end{array}$$

Agent A also looks for recommendation about b . A finds out that (i) agent D , which is already in $Trec(A)$ and whose referral trust has been evaluated 0.8, recommends a rating 0.4 for b (i.e., according to D , b is untrustworthy) (ii) agent E , which does not belong to $Trec(A)$ recommends a rating 0.8 for b (i.e., E 's opinion is that b is a trustworthy restaurant). Both agents mention the context “romantic dinner”. By considering recommendations (i.e., rule (2.a)), A can obtain two more instances $A \bullet \xrightarrow{(w_4, m)} b$. Later, A

can merge them with the instance obtained by Alice's past experience. First, A needs to evaluate its referral trust in D and E for the current time w_4 . Because $A \xrightarrow{\frac{w_3, 0.8}{O}} D \in Trec(A)$, A can “refresh” its referral trust by applying rules (5.b) and (7.b), as follows:

$$\frac{\frac{A \xrightarrow{(w_3, 0.8)} D}{O}}{A \bullet \xrightarrow{(w_4, 0.8)} D} \quad \frac{A \bullet \xrightarrow{(w_4, 0.8)} D}{O}}{A \xrightarrow{(w_4, 0.8)} D}$$

Concerning E , A estimates its referral trust by applying rule (5.a) and later (7.b). If we assume that $\text{estimate}_A(E) = 0.7$, we have:

$$\frac{\frac{[\text{estimate}_A(E) = 0.7]}{A \bullet \xrightarrow{(w_4, 0.7)} E}}{O}}{A \xrightarrow{(w_4, 0.7)} E}$$

Without an analysis of justification, and if we suppose that A only accepts recommendations coming from members whose referral trust is over the threshold $w_0 = 0.69$, A would accept both D 's and E 's recommendations. By discounting and merging the two recommendations (rules (2.a) and (4.a)), the trust in b would be $(0.8 \cdot 0.4 + 0.7 \cdot 0.8)/2 = 0.44$. This result would be merged, by applying the same rules, with the trust that A gets from Alice's past experience. This would make a final value of $(0.75 + 0.44)/2 = 0.6$. Agent A would then finalise the trust evaluation process (by rule 4.b). The final conclusion is that b is a weak trustworthy restaurant. Alice, who prefers trustworthy restaurants, would thus decide not to try b .

However, if argumentation is used, the final judgement could be different and more appropriate to Alice's profile. In this case, A evaluates the *argumentation trust* of D by asking for a justification about D 's

low rating of 0.4. Different argumentation dialogues take place between A and D , each for each aspects that D has considered when composing its rating in b . For example, the dialogue concerning the “ambiance” informally appears like what follows:

A: why did you give 0.4?
(demand a justification)

D: I didn’t like the restaurant
because the restaurant was too quiet
(ground for a justification)

A: Yes, but this is romantic, isn’t it?
(counter-argument, demand a justification)

D: No, because there were too many moments
of silence between me and my partner
(ground for the justification)

A: I see, but this is not important for me
(rebuttal of argument)

Similar dialogues clarify the point of view of A and D on two other aspects, namely, “quality of food” and “price”. Let us suppose that only D ’s justification about the “quality of food” is accepted by A . Thus, according to formula (8), the argumentation trust between A and D is $1/(3 + \frac{1}{4}) = 0.3$ and $A \xrightarrow{w_4, 0.3} D$. When this value is “added” to the referral trust of D , 0.8, the resulting ω'' is 0.55 (see rule (2.a)). Consequently, D ’s recommendation is not considered in this tournament because it is below the threshold ω_0 .

On the contrary, after arguing with E , A accepts all the arguments of E over “ambiance”, “quality of food” and “price” (we do not show the dialogues here). Agent A and E have a perfect match of opinions, and the argumentation trust calculated with the formula (8) is $3/(3 + \frac{1}{4}) = 0.9$ and $A \xrightarrow{w_4, 0.9} E$. When this value is “added” to the referral trust of E , 0.7, the resulting $\omega'' = 0.7 \oplus 0.9$ is 0.8. The recommendation of E is then accepted because it is above the threshold ω_0 , and the temporary functional trust in b is $(0.9 \cdot 0.8) = 0.72$ according to rule (2.a) applied as follows:

$$\frac{A \xrightarrow[\emptyset]{(w_4, 0.7)} E \quad E \xrightarrow[(E, 0.8); C]{(w_3, 0.8)} b \quad A \xrightarrow{w_4, 0.9} E}{A \xrightarrow[(E, 0.8); C]{w_4, 0.74} b} \quad 0.7 \oplus 0.9 > 0.69$$

Finally, when this last value is merged with the trust that A gets from Alice’s past experience, the final trust is $(0.75 + 0.72)/2 = 0.74$ (rule (4.a)):

$$\frac{A \bullet \xrightarrow[\{(A_{w_3}, 0.75)\}; C]{(w_4, 0.75)} b \quad A \bullet \xrightarrow[\{(E, 0.72)\}; C]{(w_4, 0.72)} b}{A \bullet \xrightarrow[\{(A_{w_3}, 0.75), (E, 0.72)\}; C]{(w_4, 0.74)} b} \\ \frac{A \bullet \xrightarrow[\{(A_{w_3}, 0.75), (E, 0.72)\}; C]{(w_4, 0.74)} b}{A \xrightarrow[\{(A_{w_3}, 0.75), (E, 0.72)\}; C]{(w_4, 0.74)} b}$$

In this case, A ’s suggestion is that b is trustworthy and Alice will try restaurant b .

After trying restaurant b , Alice rates it 0.9 (rule (1.c)); she liked the romantic ambiance of the restaurant. Consequently, agent A raises the referral trust of E , for example, from 0.7 to 0.75 (see rule (6.a)). In contrast, D ’s referral trust remains unchanged, because its recommendation has not been considered by A . D still can suggest nice restaurants later on.

6 AGENT-BASED SYSTEM ARCHITECTURE

In order to implement the proposed trust model, our *delegate agents* need the following capabilities (i) *Reasoning*: it should be able to evaluate trust values, build and update its knowledge ($TRat$ and $TRec$), and argue with other peers; (ii) *Autonomy*: it has to process the aforementioned tasks autonomously (without any manual assistance from its user); and (ii) *Context-awareness*: it needs to capture the context of its user, which is needed to reason about the trust.

To fulfil the aforementioned requirements, we design the VA as an *open multi-agent system* (open MAS) (Barber and Kim, 2002), which represents a scalable and flexible system that matches our virtual community concept. Moreover, the two main features of open MAS members: (i) can freely join and leave at any time and (ii) are owned by different stakeholders with different aims and objectives, perfectly fit the description of our VA’s *delegate agents*. However, our architecture is not completely decentralised. In fact, we use a central component called *Bulletin Board* which is in charge of keeping up-to-date the list of present members (useful for members discovery) and the list of items to be evaluated (see also Figure 5).

With respect to the internal architecture of *delegate agents*, we adopt a *Belief-Desire-Intention* (BDI) (Rao and Georgeff, 1995). The BDI model offers an interesting framework to design deliberative agents which are able to act and interact autonomously and according to their mental states. The main components of the *delegate agent*’s architecture are illustrated in Figure 4. Rounded rectangles represent processes while rectangles represent the different data. In the “Memory” component, two different shapes are used to show whether the data is an input (e.g., *Profile*) or an output (e.g., *Answer*). In brief, the “Goal Generator” (corresponding to *Desires* in the BDI model) produces goals that the agent has to follow. A goal could be: to answer a user’s (or peer) request, to update its own $TRat$ and $TRec$, etc. These

goals are also influenced by the “User Profile” (this includes the user’s context). In order to fulfil these *Desires* (or goals), the *delegate agent* has to formulate a set of *Intentions*, which will become actions. These *Intentions* are dictated and later executed by the “Recommender” and the “Argumentation Engine”. As a result of these actions, the knowledge (here, *TRat* and *TRec*) of the *delegate agent* is updated, which constitutes the *Beliefs* of the agent. Based on these new *Beliefs*, more *Intentions* have to be processed (if the current goal is not yet satisfied) or a new goal is set (or updated). The same cycle continues as long as there are goals to be achieved. More details about this architecture are available in (Sahli et al., 2008).

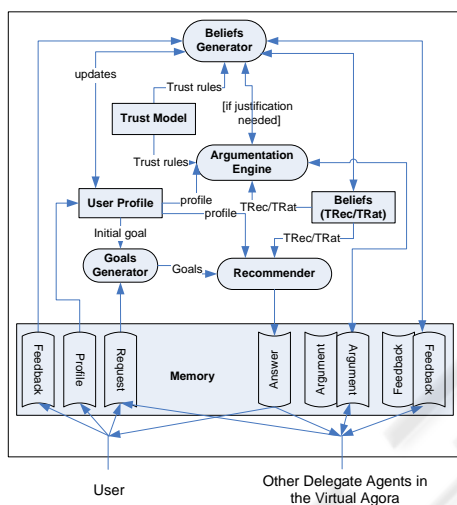


Figure 4: The *delegate agent*'s BDI architecture.

Extension to Mobile Users. We now demonstrate the easy extensibility of our architecture to support mobile users. Since *delegate agents* have to request/argue about opinions to ensure high-quality recommendations, they obviously require more interaction. However, when users are mobile, exchanging these messages between peers (mobile users) would generate a large and costly wireless communication traffic. It is thus necessary to avoid remote messages as much as possible and allow most of communication to be held locally (peers exchanging messages should be located at the same server). The VA concept seems to be appropriate to fulfil this requirement since it constitutes a meeting infrastructure where all *delegate agents* can exchange local messages. But how to make the link with mobile users? To achieve this goal, we extend our agent-based architecture by assigning a second agent (in addition to the *delegate agent*) to each user (here mobile user). We call this agent *embedded agent*. This light (has few data and

functionalities) agent is a proxy between the user and the *delegate agent*. It is embedded in the mobile device of the user. It mainly (i) notifies *delegate agent* about the user’s feedback, tags (e.g., ratings), changes of interests or preferences, etc., and (ii) requests recommendations on behalf of the user. While *delegate agent* is deliberative, embedded agent is more a reactive agent. Indeed, it does not support any reasoning, it is only making the bridge between the user and the *delegate agent* and reacting to incoming events. The architecture described previously is thus extended as shown in Figure 5. More details about the internal architecture of this agent are presented in (Sahli et al., 2008).

Implementation. We have already implemented a prototype of the VA. We chose JADEX as a development environment. Besides the fact that JADEX is built over the reliable environment Jade, it handles the BDI concept which is very useful in our case to easily implement the VA’s members. We are currently working in integrating the automatic capturing of the user’s context to the architecture. We are also working on implementing the argumentation mechanism as described in this paper.

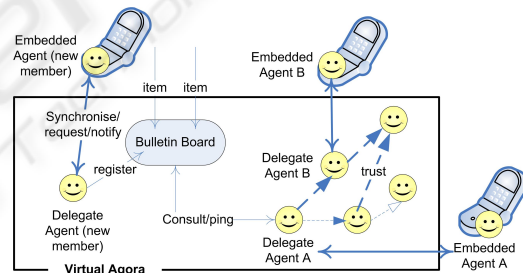


Figure 5: Simplified architecture supporting mobile users.

7 RELATED WORK

Very few papers have addressed the use of argumentation in their trust models. Stranders (Stranders et al., 2007) seems to be the first to use a form of argumentation for trust evaluation viewed as a decision process, taking inspiration from (Amgoud and Prade, 2004). Prade proposed a similar approach in (Prade, 2007). However, in these systems, argumentation is an internal process (internal to each agent) used to support decision making. Each agent evaluates its trust in a peer (or a source) according to a set of arguments. In contrast, in our approach, each agent uses argumentation as a negotiation mechanism with other agents in order to debate about the “strength” of a given item. To our knowledge, only Bentahar and

Meyer (Bentahar and Meyer, 2007) have addressed an argumentation-based negotiation to enhance the trust model of agents. Their approach is, by the way, complementary to ours. First the goal is to provide a secure environment for agent negotiation within multi-agent systems. Second, the overall trust degree of an agent is estimated according to a probabilistic model which depends upon the number of past and current accepted arguments.

Our work is also related to numerous works that have been addressing distributed trust management in the last years. Our solution of making trust dependent of past experience and recommendations is a common choice in distributed trust management (e.g., see the surveys (Ruohomaa and Kutvonen, 2005; Jøsang et al., 2005)).

Our design is also related to recommender systems and collaborative filter solutions. The abstract functions we use to initiate trust (i.e., *estimate*, *eval*) and to tag items (i.e., *tag*) find their concrete counterpart in the solutions that exist to estimate the trustworthiness of objects during the bootstrap phase of trust. Again a plethora of research works is available in the literature. Quercia *et al.* propose to use a user's local information (on the ratings of its past experiences) to estimate the trustworthiness of an item (Quercia et al., 2007b). In (Quercia et al., 2007a), they also design a distributed algorithm that users can run to predict their referral trust in content producers from whom they have never received content before. Both the solutions are suitable for mobile devices, which make them attractive in our design.

Argumentation has been proposed to enhance critic systems and recommender systems (Chesnevar et al., 2006). Potential suggestions which follow a user's web query, are ordered by relevance. The relevance relation is defined in terms of argumentation, whose rules express the user preferences. Warranted suggestions come first, then undecided followed by defeated suggestions. Differently from our approach, *Defeasible Logic* is used to express a user's preferences, which can be quite naturally expressed as a defeasible reasoning. In this case, argumentation is not a dialogue between two subjects (as in our approach) but it is the result of attacks/defenses pairs that arise in the subjective defeasible reasoning of a user when she/he evaluate a piece of information retrieved from the web. Argumentation has been also proposed in agent systems to reason about desires of agents (Rotstein et al., 2007). Here, *Defeasible Logic* is used to select, among a set of potentially conflicting agent's desires, an appropriate one that fits the particular situation of a given agent.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we enhance the decentralised recommender system we proposed in (Lenzini et al., 2008) by integrating argumentation into the trust model. Within a virtual community of raters, a member can thus select recommendations not only based on its trust in these peers, but also on the justifications they give. We thus make a member take advantage of recommendations coming from less trustworthy members (e.g., those with which she/he has no direct experience) if they are strongly justified by convincing arguments. As future work, we intend to propose a full implementation of the argumentation model. This implies finding a way to capture the user's way of reasoning (in a user-friendly manner) and integrating the output in the formal model. An eventual commercialisation of our recommender system will thus depend on linking the formal model with the real data (captured from the user) via a user-friendly interface.

With respect to the supporting architecture, our decentralised recommender system is designed based on the agent paradigm. In fact, we model raters by autonomous and unobtrusive agents, which frees users from managing their trust in other members. Moreover, if users are mobile, we decouple each agent into two agents (*delegate* and *embedded*), which dramatically reduces wireless traffic between peers. We are currently working on integrating *ContextWatcher* (Koolwaaij et al., 2006) into our system in order to effectively support the context. The *ContextWatcher* is a mobile application developed in our research laboratory, and which aims at making it easy for an end-user to automatically record, store, and use context information. It automatically captures some context aspects such as time, location, and social environment (of the end-user).

REFERENCES

- Amgoud, L. and Prade, H. (2004). Using arguments for making decisions: a possibilistic logic approach. In *Proc. of the 20th conference on Uncertainty in artificial intelligence AUAI'04, Banff, Canada*, pages 10–17. AUAI Press.
- Barber, K. S. and Kim, J. (2002). Soft security: Isolating unreliable agents from society. In *Proc. of the 5th Workshop on Deception, Fraud and Trust in Agent Societies, AAMAS 2002, July 15-19, Bologna, Italy*, pages 8–17. ACM.
- Bentahar, J. and Meyer, J. J. C. (2007). A new quantitative trust model for negotiating agents using argumentation. *Int. Journal of Computer Science & Applications*, 4(2):39–50.

- Chesnevar, C. I., Maguitman, A. G., and Simari, G. R. (2006). Argument-based critics and recommenders: a qualitative perspective on user support systems. *Data Knowl. Eng.*, 59(2):293–319.
- Garcia, A. and Simari, G. R. (2002). Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138.
- Jøsang, A., Gray, L., and Kinateder, M. (2006). Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems Journal*, 4(2):139–161.
- Jøsang, A., Ismail, R., and Boyd, C. (2005). A survey of trust and reputation systems for online service provision. *Decision Support Systems*. (available on line on ScienceDirect) in press.
- Koolwaaij, J., Tarlano, A., Luther, M., Nurmi, P., Mrohs, B., Battestini, A., and Vaidya, R. (2006). Contextwatcher - sharing context information in everyday life. In *Proc. of the IASTED Int. Conf. on Web Technologies, Applications, and Services (WTAS2006)*, Calgary, CA, pages 39–60. ACTA Press.
- Lenzini, G., Sahli, N., and Eertink, H. (2008). Agents selecting trustworthy recommenders in mobile virtual communities. In *Proc. of the 11th Int. Workshop on Trust in Agent Societies (TRUST08)*, AAMAS 2008, pages 94–104.
- Miller, B., Konstan, J., and Riedl, J. (2004). Toward a personal recommender system. *ACM Transactions on Information Systems*, 22(3):437–476.
- Nute, D. (1994). Defeasible logic. In Gabbay, D., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 353–395. Oxford University Press, Oxford.
- Prade, H. (2007). A qualitative bipolar argumentative view of trust. In Prade, H. and Subrahmanian, V. S., editors, *Proc. of the Int. Conf. on Scalable Uncertainty Management (SUM)*, Washington, 10-12 Oct 2007, DC, USA, volume 4472 of *LNAI*, pages 268–276. Springer-Verlag.
- Prakken, H. (2006). Formal systems for persuasion dialogue. *Knowledge Engineering Review*, 21(2):163–188.
- Preece, J. (2000). *Online Communities: Designing Usability, Supporting Sociability*. John Wiley and Sons.
- Quercia, D., Hailes, S., and Capra, L. (2007a). Lightweight distributed trust propagation. In *Proc. of the 7th IEEE International Conference on Data Mining, 28-31 Oct 2007, Omaha, NE, USA*, pages 282–291. IEEE Computer Society.
- Quercia, D., Hailes, S., and Capra, L. (2007b). TRULLO - local trust bootstrapping for ubiquitous devices. In *Proc. of the 4th IEEE International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, Philadelphia, US.
- Rao, A. S. and Georgeff, M. (1995). Bdi agents : from theory to practice. In *Proc. of the 1st Int. Conference on Multi-Agent Systems (ICMAS95)*, June 12-14, San Francisco, USA, pages 312–319. AAAI Press.
- Rotstein, N. D., García, A. J., and Simari, G. (2007). Reasoning from desires to intentions, a dialectical framework. In *Proc. of the 22nd. AAAI Conference on Artificial Intelligence (AAAI 2007)*, Vancouver, Canada, 22-26 July 2007, pages 136–141. AAAI Press.
- Ruohomaa, S. and Kutvonen, L. (2005). Trust management survey. In *Proc. of the iTrust 3rd Int. Conference on Trust Management, 23-26, May, 2005, Rocquencourt, France*, volume 3477 of *LNCS*, pages 77–92. Springer-Verlag.
- Sabater, J. and Sierra, C. (2002). Social regret, a reputation model based on social relations. *SIGecom Exchanges*, 3(1):44–56.
- Sahli, N., Lenzini, G., and Eertink, H. (2008). Trustworthy agent-based recommender system in a mobile p2p environment. In *Proc. of the 7th Int. Workshop on Agents and Peer-to-Peer Computing (AP2PC08)*, AAMAS 2008, pages 1–11.
- Stranders, R., de Weerd, M., and Witteveen, C. (2007). Fuzzy argumentation for trust. In Sadri, F. and Satoh, K., editors, *Proc. of the 8th Work. on Computational Logic in Multi-Agent Systems (CLIMA VIII)*, 10-11 Sep 2007, Porto, Portugal, *LNAI*. Springer-Verlag. (in press).
- Teacy, W., Patel, J., Jennings, N., and Luck, M. (2005). Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *Proc. of the 4th Int. Joint Conf. AAMAS*, pages 997–1004.
- Toivonen, S., Lenzini, G., and Uusitalo, I. (2006). Context-aware trustworthiness evaluation with indirect knowledge. In *Proc. of the 2nd Semantics Web Policy Workshop (SWPW 2006)*, CEUR Workshop Proceedings. CEUR-WS.org.