Towards a Generic Trust Model – Comparison of Various Trust Update Algorithms

Michael Kinateder*, Ernesto Baschny, and Kurt Rothermel

Institute of Parallel and Distributed Systems (IPVS)
Universität Stuttgart
Universitätsstr. 38
70569 Stuttgart, Germany

{kinateder, rothermel}@informatik.uni-stuttgart.de, ernst@baschny.de

Abstract. Research in the area of trust and reputation systems has put a lot of effort in developing various trust models and associated trust update algorithms that support users or their agents with different behavioral profiles. While each work on its own is particularly well suited for a certain user group, it is crucial for users employing different trust representations to have a common understanding about the meaning of a given trust statement.
The contributions of this paper are three-fold: Firstly we present the UniTEC generic trust model that provides a common trust representation for the class of trust update algorithms based on experiences. Secondly, we show how several well-known representative trust-update algorithms can easily be plugged into the UniTEC system, how the mappings between the generic trust model and the algorithm-specific trust models are performed, and most importantly, how our abstraction from algorithm-specific details in the generic trust model enables users using different algorithms to interact with each other and to exchange trust statements. Thirdly we present the results of our comparative evaluation of various trust update algorithms under a selection of test scenarios.

1 Introduction

The phenomenal growth of the Internet that we experienced during the last couple of decades, together with the fact that computers can be found not only in business environments but also in many households almost to the point of being a commodity nowadays, led to a widespread public acceptance of this medium. There are plenty of reasons why people connect to the Internet. Among the most common usage scenarios are getting *access to information*, *communicating with people* and *buying or selling* goods.

There is no doubt that the Internet offers masses of information in all kinds of different areas, ranging from purely leisure-relevant possibly dispensable information, like who is currently number one in the US-single-charts, to more critical areas, like product reviews or even stock exchange data. Especially in these critical areas, the user needs *correct* information. Therefore the user needs to decide whether the information provider is trustworthy or not. In real life, we use social network structures of friends, colleagues etc. to find trustworthy persons whom to get advice or general information from. In the virtual environment of the Internet, *reputation systems* model these structures up to a certain degree supporting users in their decision whom to trust and whom to avoid. The goal of these systems is to minimize the risk of interactions with strangers.

One aspect, that research in trust and reputation systems strives to determine, is a suitable digital representation of trust, commonly referred to as a *trust model*. Tightly interwoven with trust models are the algorithms used to determine, how this trust is updated according to different usually discrete events. Such events might be a new experience with the person in question, or new information from other trusted sources regarding the reputation of this person etc. Numerous different models and trust update algorithms have been proposed in the literature and each approach is particularly well suited for a certain user group or application area. However, these trust models are not interoperable since there is a lack of a generic representation of trust. A generic trust model would allow users intending to use different models to translate their local representation to the generic one in order to understand each other's trust statements.

Our contribution is built on the observation that, although the algorithms used to compute a certain trust value are quite different from each other, the data that the algorithms are working upon and the outcome of the algorithms are not that different and can thus be mapped on a generic model. We suggest one approach for such a generic representation which we implemented in the context of the UniTEC distributed reputation system. This generic trust model allows us to easily integrate various existing trust update algorithms. Another contribution lies in a comparative analysis of these algorithms, which presents how the algorithms react on various test scenarios. This has – according to our knowledge – not been done in this depth before.

We structure our paper as follows: In the next section we give a brief overview of the UniTEC reputation system, in whose context this research is being conducted. After discussing several general aspects of trust and trust relationships in Sect. 3 we present in detail the components of the generic trust model in Sect. 4. We introduce in Sect. 5 a subset of trust update algorithms implemented in UniTEC and the necessary adaptations. In Sect. 6 we describe several test scenarios that the algorithms are subjected to which is followed in Sect. 7 by a presentation of the results of this evaluation. We conclude our paper in Sect. 8.

2 Application Area for a Generic Trust Model

In this section, we briefly point out the functionality of the UniTEC system as one sample application area for the introduced generic trust model. UniTEC is a completely decentralized reputation system and consists of a peer-to-peer network of agents residing on nodes with communication capabilities.

For privacy reasons, each user employs multiple virtual identities or pseudonyms instead of his real identity when interacting with the UniTEC system. Each pseudonym has an associated public and private key pair and is responsible for one or more context areas (see Sect. 3). The *identity management component* allows to create or remove pseudonyms and to assign context area responsibilities. The *anonymous peer-to-peer (P2P) communication component* provides communication mechanisms between pseudonyms while protecting the link between real user identities and their pseudonyms (see also [1]).

UniTEC can store and request trusted data items (TDI) about arbitrary products or services. TDIs are recommendations digitally signed with the key of the appropriate pseudonym and stored in the XML database of the *data management component* of the pseudonym owner's UniTEC agent. In order to retrieve a TDI, a requesting user poses a query to its own agent, which determines from the query context a neighborhood of already known pseudonyms deemed as capable of answering the query. The query is disseminated through the means offered by the anonymous P2P communication component to each neighborhood member and from there recursively further. During the query dissemination, a construct called the *trust chain* is built as part of the query, which consists of a set of trust statements, each specifying the trust of a node in its successor starting with the original requester. A node which has stored a TDI that satisfies the query sends a TDI response to the requester that contains this TDI and the then completed trust chain.

The *trust management component (TMC)* evaluates the trust chains contained in the TDI responses and presents to the requester the TDIs together with the calculated transitive trust in the TDI-issuer. Furthermore, the TMC keeps track of the user's trust in each pseudonym that she or he has been in contact with. More concretely, it stores trust in its database according to the specified trust model and updates the trust in these pseudonyms upon receipt of user feedback regarding the quality of the received TDIs according to the trust update algorithm specified in the user's preferences. This trust update influences the neighborhood selection the next time that a query is received for the trust context in question.

A generic representation of trust is essential especially for the trust statements inside the trust chains to enable the requester's TMC to compute the transitive trust in each TDI-issuer independently from the local trust models used at each intermediary. We are well aware of the fact, that this brief introduction leaves many questions unanswered. For more in-depth information

regarding UniTEC, we would like to point the interested reader to [2] and our project website[1].

After having presented background information regarding the UniTEC reputation system as a whole, we focus in the following on the capabilities of its trust management component TMC. We start by introducing our view on the various aspects of a trust relationship.

3 The Phenomenon of Trust Relationships

In order to understand the phenomenon of trust relationships, we first need to understand the meaning of *trust*. In the related work, various different definitions of the term trust have been proposed. One definition popular in the agent field is from Diego Gambetta [3] "trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action...". We identify two relevant points in this definition: firstly, trust is used in order to *predict* an entity's *future behavior*, secondly, trust is *subjective*. The subjectivity leads consequently to asymmetric trust relationships between *trustor*, the entity who is trusting, and *trustee*, the entity who is trusted. In the following, we identify various *dimensions of trust relationships* in addition to trustor and trustee:

Trust measure refers to the quality of the trust relationship, which ranges from complete distrust over a neutral trust measure to full trust. The more a trustee is trusted, the higher the trust measure is supposed to be.

Trust certainty specifies the confidence of the trustor in his or her estimation of the trustee. If this estimation is gained via only few personal experiences or just via word of mouth, the certainty is supposed to be low.

Trust context People trust in a fine-grained manner depending on the area and goal in question, for instance person A might trust person B to babysit her child whereas she might not trust person B to repair a computer. A context can be represented by different categories as we described in [2].

Trust directness Direct and indirect trust [4,5] represent two distinct trust relationships. Direct trust means that the trustee can directly cooperate with the trustor. With indirect trust, the trustee is not supposed to cooperate directly himself, but should forward the cooperation request to a good expert. Consider for instance person A knowing that person B has many friends working in the computer business, although B is not schooled in this context herself. A will not trust B directly with a repair task but might very well trust recommendations received indirectly via B from one of B's expert friends.

Trust dynamics A trust relationship is not static, but changes dynamically on various different incidents, e.g. on *own direct experiences*. If for instance the babysitting of A's child by B went well, the trust of A in B will increase. In addition to own experiences, *trust estimations received from others* influence the own trust assessment as well. If A's good friends C, D, and E warn

[1] `unitec.informatik.uni-stuttgart.de`

A about the unreliable nature of B, A might refrain from relying on B's babysitting capabilities. Lastly, quite interestingly, *trust relationships may also change over time* when no experiences have been made, a fact, that is up to our knowledge not covered in the related work yet.

4 Towards a Generic Trust Model

Having presented the general concepts of trust relationships in the previous section, we describe in the following, how these concepts are mapped on the components of our generic trust model. The key components of our model result from an analysis of the characteristics of various existing trust models.

4.1 Trust Measure and Certainty

Various different representations of trust values exist in the related work. Trust values can be depicted as real numbers in certain intervals like for instance $[-1, +1]$, as done by Jonker and Treur [6] and Sabater [7] or probabilities in $[0, 1]$, as proposed among others by Jøsang and Ismail [8], Yu and Singh [9], and Kinateder and Rothermel [2]. Others propose discrete values, like the binary representation by Blaze and Feigenbaum [10] or four discrete values introduced by Abdul-Rahman and Hailes [5].

The metric used for the *trust measure* in our proposed generic trust model is a real number in the interval of $[0, 1]$. Complete distrust is represented by 0 whereas 1 corresponds to full trust. This representation allows an easy transformation of any previously described measures in the generic measure as we will see in more detail in Sect. 5.

Not all investigated algorithms support the computation of a certainty value, which states the quality of the trust assessment represented in the trust measure. If uncertainty is mentioned [9,8,7] it is specified in the interval $[0, 1]$.

The *trust certainty* in the generic trust model is represented similarly to the trust measure as a number in the interval of $[0, 1]$, whereas 0 describes complete uncertainty and 1 the total certainty.

4.2 Trust Context

As pointed out in the previous section, applications can define various context areas in which entities can be trusted. It is important to note, that these areas are not necessarily independent from each other. Different kinds of dependencies can exist among the context areas: *instance-of* relationships are one-level relationships for *classification*, *is-a* relationships provide *generalization*, *part-of* relationships enable *aggregation* and surely many other – potentially application-specific – forms of dependencies between context areas can be imagined.

For the sake of the trust model, however, we do not need to model all these relationships in detail. Instead, we model the *asymmetric semantic distance* between the context areas and therefore abstract from the kind of dependency.

The metrics chosen for the semantic distance is a real number in the interval of $[0, 1[$. A distance close to 1 represents a high dependency, a distance of 0 refers to no dependency. Therefore, we organize the trust context areas as a weighted directed graph as can be seen in 1. This allows us to spread the impact of a trust update in one area to related areas. The context areas and the semantic distances between the areas are specified by the applications to be supported with trust management. However, due to the subjectivity of trust, each user is enabled to locally modify the distances to suit his or her personal views.

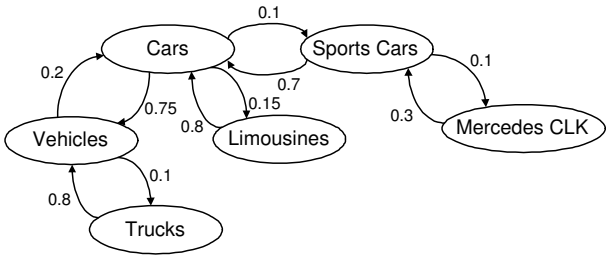

Fig. 1. Example snippet of a weighted directed trust context graph. The weights represent sample semantic distance of the context areas.

4.3 Trust Directness

Another dimension of a trust relationship is its directness. In our model, direct and indirect trust are two distinct instances, each with a specific trust measure, certainty etc. They are stored and updated separately by the trust algorithms.

4.4 Trust Dynamics

As already mentioned in the introduction, a change in trust occurs upon receipt of feedback regarding an experience of a trustor with a trustee. Various aspects are discussed in the following that influence the trust dynamics.

Quality Feedback The trustor provides feedback about the subjective quality of a received information item. The metrics used to rate the quality is a real number in the interval of $[0, 1]$. A perfect information item is rated with 1, 0 describes a completely unsatisfactory one. The generic trust model does not dictate how this feedback is gained; e.g. for recommendations of a static attribute-value structure this feedback can be gained automatically in a collaborative filtering style.

Trustor Confidence Trustors may specify a confidence in their own offered information items. This confidence is represented by a real number in the interval of $[0, 1]$. Similar to the trust certainty, 0 stands for no confidence whereas 1 stands

for the highest possible confidence in the offered information. This confidence influences the trust update such that a weak statement with a low confidence leads to only a slight trust update, whether positive or negative.

Transaction Utility Each information request, and the corresponding responses and feedback statements refers to a certain transaction the requester or trustor is about to take. Depending on the transaction's significance, the trustor specifies the utility as a real number in the interval of $[0, 1]$. We assume, that a "maximum utility" can be specified in such as utilities higher as this maximum utility will lead to the same trust update impact as with the specified maximum utility. 1 refers to the normalized maximum utility which leads to a trust update with a higher impact.

Experience Aging (Optional) The quality of trustees is not necessarily constant but may change over time, for instance due to gathered experience in a certain field. In order to determine trust as a prediction of the future behavior, it is possible to specify, that the latest experiences ought to weigh more than older experiences. We propose two options for experience aging: a *feedback window* and an *experience aging factor*. The feedback window limits the amount of considered experiences, either depending on a certain number of experiences or a certain maximum age. The aging factor in the interval of $[0, 1]$ determines the ratio of a new experience to previous experiences in the update computation. We describe in the following section how this aging factor is used in the algorithms.

Related Trust Context Areas (Optional) As mentioned before, an update in a single trust context area A may lead to an update of a lesser extent in related areas B_i according to the relationships in the context area graph. The semantic distance between two context areas that are linked via one or more intermediary areas can be computed by calculating the product of the semantic distances along the path. The proportion of the update of B_i to A is determined by the strongest semantic distance from A to B_i, in other words by calculating the maximum product of all paths from A to B_i. Context area that cannot be reached from A or where the distance is not known are not updated.

Trust Fading (Optional) When no experience with a trustee is made in a long time, the old trust relationship might no longer be valid. This usually means that the trust confidence level decreases over time. But there might be situations or time frames when also the trust level decreases without new experiences. We represent the magnitude of this fading effect with a *fading factor* as a real number $\lambda \geq 0$. A factor of 0 means no fading effect. The higher the fading factor, the faster trust relationship drops back to a state specified by the trust algorithm. This state might be a state of no trust and no confidence.

5 Supported Algorithms and Necessary Adaptations

The generic trust model presented in the previous section was conceived in such a way that existing trust models could be easily integrated into UniTEC. In the

following, we present the mapping of the local trust models to the introduced generic model and suggest some algorithmic adaptations. The subset of investigated algorithms discussed here are Abdul-Rahman–Hailes, Beta Reputation, ReGreT and the original UniTEC algorithm. Due to space constraints, our results on the work of Yu and Singh [9], their previous suggestions [11] and Lik Mui's algorithm [12] are not covered here.

5.1 Abdul-Rahman – Hailes

The work on a trust model in [5] is based on sociological studies similar to the work of Marsh [13]. Here, interpersonal trust is context-dependent, subjective and based on prior experiences. A reputation information exchange amongst members of the community assists on trust decisions. All these aspects fit well in our generic trust model.

Trust is measured in a discrete metric with four values: very untrustworthy *vu*, untrustworthy *u*, trustworthy *t* and very trustworthy *vt*. Abdul-Rahman and Hailes describe three uncertainty states, which complement the four trust values: more positive experiences u^+, more negative experiences u^-, and equal amount of positive and negative experiences u^0. Ratings are specified in a discrete metric: very bad *vb*, bad *b*, good *g*, very good *vg*.

To fit this into the generic trust model, the discrete trust values have to be mapped onto the scalar trust metric. The range $[0, 1]$ is split into three equally sized ranges. The values at the borders of these ranges represent the four values from the discrete metric $(0, \frac{1}{3}, \frac{2}{3}, 1)$. Each discrete value is assigned a single position number ($pos(0) = vu$, $pos(1) = u$, $pos(2) = t$, $pos(3) = vt$). To map a value from the generic trust model (t_g) onto this discrete metric (t_d), the following calculation is applied: $t_d = pos(round(t_g \cdot 3))$. The same formulas are applied for mapping the four discrete rating values. This translation follows the reasoning that trust in this model cannot be greater than *very trustworthy*, which is represented by the value of 1 in the generic trust model.

The semantics of the uncertainty values is not defined in [5], therefore the mapping into our generic trust model is difficult. For the four trust values, no uncertainty is known, which is represented as a certainty of 1 in our generic trust model. The initial trust value (when no previous experiences are known) is represented by the u^0 uncertainty value, so it makes sense to keep the generic certainty value of 0 (the generic trust value is of no importance in this case, so we also keep it at the lowest level of 0). Rahman's paper does not give an explanation on how to interpret this uncertainty value in other situations. The uncertainty values u^+ and u^- represent states where slightly more positive (or negative) previous experiences have been recorded. This is expressed in our generic trust model by a slight mistrust (1/3) or a small positive trust (2/3). In these cases the uncertainty component is represented by a mean generic certainty value (0.5).

5.2 The Beta Reputation System

The Beta Reputation System [8] is based on Bayesian probability. The *posteriori* probability of future positive experience is represented as a beta distribution based on past experiences. The trust value, in this work called "reputation rating", is determined by the expectation value of the corresponding beta distribution. This is a probability value in the scalar range $[0, 1]$. A one-to-one mapping to our generic trust value is possible. The certainty of the trust calculation is defined in this paper by mapping the beta distribution to an opinion, which describes beliefs about the truth of statements ([14,15]). In this mapping the certainty starts at 0 and grows continuously to 1 with more experiences being considered. This metric also can be directly mapped to our scalar generic certainty metric $[0, 1]$. Experiences in the Beta Reputation System are rated through two values: $r \geq 0$ for positive evidence and $s \geq 0$ for negative evidence. The sum $r + s$ represents the weight of the experience itself. These two weighted rating values can be mapped to the generic rating value ($0 \leq R \leq 1$) and the generic weighting metric ($0 \leq w \leq 1$) as $r = w \cdot R$ and $s = w \cdot (1 - R)$.

In this trust model, the accumulation of ratings can make use of a forgetting factor, which is the equivalent to the generic aging factor. In the Beta Reputation System the forgetting factor (λ_{beta}) has a reversed meaning: $\lambda_{beta} = 1$ is equivalent of having no forgetting factor and $\lambda_{beta} = 0$ means a total aging (only the last experience counts). Thus $\alpha = 1 - \lambda_{beta}$ represents a simple mapping to our generic aging factor.

5.3 The ReGreT System

The ReGreT system [7] represents a reputation system which uses direct experiences, witness reputation and analysis of the social network where the subject is embedded to calculate trust.

Direct experiences are recorded as a scalar metric in the range $[-1, +1]$. Trust is calculated as a weighted average of these experiences and uses the same value range. A mapping to the generic values can be done by transforming these ranges to $[0, 1]$ (shifting and scaling). A reliability is calculated for each trust value, based on the number of outcomes and the variation of their values. This reliability is expressed as a value in the range $[0, 1]$ which directly matches the representation of our generic certainty value.

An aging factor is not used. Instead, the oldest experience is neglected ($w = 0$), the newest experience is fully weighted ($w = 1$). The weight of experiences in between grows linearly from 0 to 1.

5.4 The Original UniTEC Algorithm

In the first work on UniTEC [2], a trust update algorithm describes the trust dynamics. It calculates a new trust value based on the old trust value and the new rating. *Ratings* in the original UniTEC proposal are expressed as a binary metric of $\{0, 1\}$ (either bad or good experience). The trust update algorithm

works as well with ratings in a scalar range of $[0, 1]$ instead, which then require no further mapping to the rating metrics of the generic trust model.

In UniTEC we specified the certainty of the trust assertion through a confidence vector, where the amount of direct and indirect experiences and a trail of the latest n direct experiences is recorded. A semantic interpretation of this vector was not given. We need to calculate the certainty as a single scalar metric as in the generic trust model. This can be accomplished in a similar manner as in the ReGreT System, where the number of experiences and the variability of its values are consolidated into a single value in the range $[0, 1]$.

We created a simple fading algorithm that works with the UniTEC update algorithm and uses the fading factor λ. In the time when no experiences are recorded, trust will linearly drop to the minimal trust value in $1/\lambda$ time units.

6 Test Scenarios

To assess the quality of the trust update algorithms presented in the previous section, a series of test scenarios was developed. Each scenario simulates a different behavior pattern of trustor and trustee as a list of ratings. This pattern is then reflected by each single trust algorithm as trust dynamics. A test scenario can bring forward a specific feature or a malfunction of a trust update algorithm.

As the test scenarios simulate the behavior of real-world people, there are certain expectations associated with the trust dynamics. A trust algorithm is expected to generate trust dynamics that satisfy these expectations. Failing to comply with the specified expectations can either be a consequence of the calculations themselves or it reflects a shortcoming of the adaptations and mappings necessary for the local trust model to work in the generic trust model.

We want to stress the fact, that due to the subjectiveness of trust in general, also the quality estimation of the behavior reflected in the trust dynamics is subjective. Therefore, we do not offer a ranking of trust update algorithms, but instead point out the distinctive features of the algorithms, so that each user can choose the algorithm that most closely reflects his own expected behavior.

OnlyMaximalRatings Starting from the initial trust state, only maximal ratings ($= 1$) are given. We would expect the trust to grow continuously and approach the maximal trust value ($= 1$).

OnlyMinimalRatings Starting from the initial trust, only minimal ratings ($= 0$) are given. If the initial trust is the minimal trust value ($= 0$), then trust should stay at this level. Otherwise, we would expect the trust value to decrease and eventually approach the minimal trust value.

MinimalThenMaximalRatings First, a series of minimal ratings is given, which is followed by a series of maximal ratings. We would expect the trust dynamics to start as described in the test scenario *OnlyMinimalRatings*. After switching to maximal ratings, trust should rise again. The expected growth rate of trust after the start of the maximal ratings should be lower than in the *OnlyMaximalRatings* test scenario.

MaximalThenMinimalRatings First, a series of maximal ratings is given, followed by a series of minimal ratings. We expect trust to rise as in the test scenario *OnlyMaximalRatings*. When the series of minimal ratings starts, trust should decrease again. The trust decrease rate in the second half of the test should be slower than in the test scenario *OnlyMinimalRatings*.

SpecificRatings After the previous test scenarios, which work with extreme ratings, these four test scenarios make use of a specific set of ratings (the *SpecificRatings*) which simulate a real-world rating situation. The ratings are: 1.0, 0.8, 0.5, 0.4, 0.5, 1.0, 0.6, 0.7, 0.7, 0.8, 1.0, 0.4, 0.3, 0.2, 0.2, 0.5, 1.0, 0.3, 0.4, 0.3. These ratings are submitted in this original order (-*Normal*), in a reversed order (-*Reversed*), ordered by ascending (-*OrderedAsc*) and by descending rating value (-*OrderedDesc*). In the *Normal* order, there are more positive ratings in the first half of the rating sequence, whereas negative ratings predominate in the second half. The expectation is that the final trust value is slightly below the mean trust value ($= 0.5$). With the *Reversed* order the expectation for the final trust value is a value slightly above the mean trust value. If the ending trust values in all four scenarios are equal, it suggests that the trust algorithm uses an *indistinguishable past* (see [6]), which means that the order of previous experiences does not matter. This should not the case when using an aging factor.

KeepPositive This scenario has a *dynamic* nature, in that it actively reacts on the resulting trust values after each individual rating. Maximal ratings are given until a certain level of trust is reached (> 0.8). This trust level is then "misused" in form of minimal ratings, until the trust value reaches a mistrust level (< 0.5). Then, maximal ratings are submitted to raise trust again. This process is repeated four times. Here the trust algorithm's reaction to attempts of misuse is analyzed. We would expect trust to quickly drop to a mistrust level when the minimal ratings occur. The *optimal algorithm* should quickly detect this misuse attempt and react appropriately, e.g. by reporting minimal trust or even blacklisting the user.

7 Evaluation

We subjected each trust update algorithm discussed in Sec. 5 with a variation of aging factors to each test scenario from the previous section. For each evaluation graph we use the representation of the algorithms presented in Fig. 2.

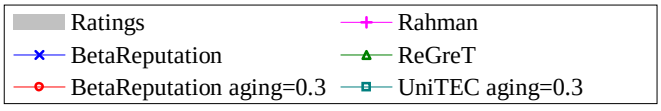

Fig. 2. Key for the trust dynamics presented in the evaluation.

Test scenario *OnlyMaximalRatings* presented in Fig. 3 illustrates the different initial trust values of the algorithms: The trust dynamics start either with a trust value of 0 (UniTEC and Abdul-Rahman) or 0.5 (Beta Reputation and ReGreT). Trust rises monotonously for all algorithms. Trust in Beta Reputation with no aging factor and UniTEC approaches asymptotically the maximum trust value. Beta Reputation with an aging factor approaches a certain level of positive trust value. ReGreT and Abdul-Rahman reach maximum trust after just one maximal rating and remain at this level.

Similar effects can be noticed in the test scenario *OnlyMinimalRatings* (see Fig. 4). The trust algorithms that started with the lowest trust value (UniTEC and Abdul-Rahman) stay at this minimum trust level. ReGreT that started at 0.5 drops to the lowest trust value after just one bad experience. Beta Reputation also started with a trust value of 0.5. Without aging factor, trust approaches asymptotically the minimum trust value. With an aging factor, trust never drops below a certain level of mistrust.

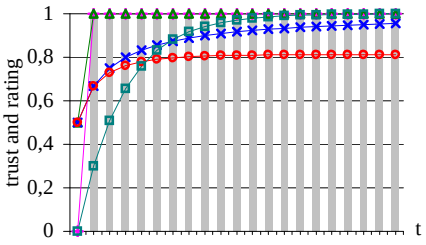

Fig. 3. OnlyMaximalRatings

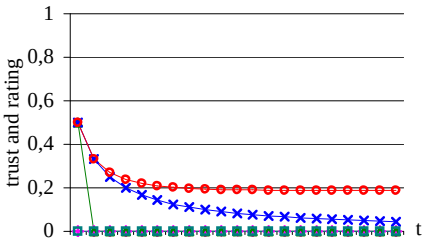

Fig. 4. OnlyMinimalRatings

Beta Reputation with an aging factor uses only a limited interval of the totally available trust value scope. This happens because the accumulation of the evidence (r and s) using a positive aging factor represents a geometric series. An upper limit for r and s thus limits the possibly reachable maximum and minimum trust values. One possible solution to make use of the whole range regardless of an aging factor is to scale the possible output range to the whole generic trust value range. This can only be done if the aging factor remains constant throughout the relevant rating history.

In *MinimalThenMaximalRatings* (Fig. 5) when the maximal ratings start, trust starts rising again in all analyzed algorithms but Abdul-Rahman's. In this latter case, trust remains at the lowest level until as much maximal ratings as minimal ratings have been received. The discrete metrics of this trust model does not support other intermediate states. Another interesting observation is that UniTEC shows the same rise on trust as in the *OnlyMaximalRatings* test scenario (rising above 0.8 after 5 maximum ratings). The other algorithms show a slower rise of trust, as we would expect after the negative impact of the negative ratings. This demonstrates one deficiency of algorithms like the original UniTEC one,

which rely solely on the last trust value and the new rating for their calculations and do not consider adequately the remaining history of ratings.

The *MaximalThenMinimalRatings* test scenario (Fig. 6) shows similar results as the previous scenario. It starts as expected like the *OnlyMaximalRatings*. When the minimal ratings start, trust drops with all but Abdul-Rahman's algorithm. Here, trust suddenly drops from maximum to the minimal value at the end of the scenario which is the point when more minimal than maximal ratings are recorded in the history.

In both scenarios we notice that Beta Reputation without an aging factor shows a slow reaction to the pattern change in the ratings.

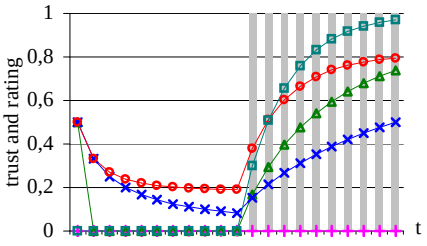

Fig. 5. MinimalThenMaximal

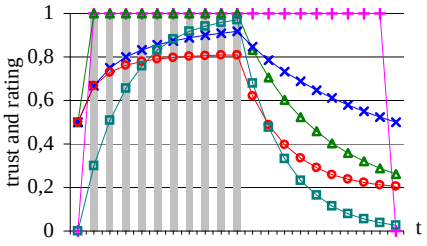

Fig. 6. MaximalThenMinimal

The *SpecificRatings* test scenarios are depicted in Figs. 7, 8, 9 and 10. In *SpecificRatingsNormal* most algorithms follow the expected trust dynamics. The ending trust value for UniTEC and Beta Reputation with aging factor of 0.3 is just below the average trust value of 0.5. ReGreT and Beta Reputation without an aging factor are a bit more optimistic ending just above 0.5. In *SpecificRatingsReversed* the opposite can be seen: The ending trust value is just above the trust value mark of 0.5.

In those four test scenarios Abdul-Rahman generates a trust dynamic that follows our expectations up until the end, when suddenly trust and certainty drop back to the lowest values. What happened here is that the algorithm reached the uncertainty state u^0. The most evident problem with this can be seen in Fig. 9. At the last couple of ratings this state of uncertainty is reached, which is not expected at all. The weakness lies in the lack of semantical meaning of the u^0 state. The only solution would be to alter the original algorithm and its underlying trust model to improve the way uncertainty is handled.

We see the characteristic of *indistinguishable past* with Abdul-Rahman and Beta Reputation without an aging factor: The ending trust values are the same regardless of the ordering of the ratings. All remaining algorithms use aging of ratings, leading to different ending values depending on the order of the ratings.

In our last test scenario, *KeepPositive*, the rating history depends on the calculated trust values. In Fig. 11 the reaction of the Beta Reputation algorithm to the scenario shows that the use of an aging factor helps with a fast reaction

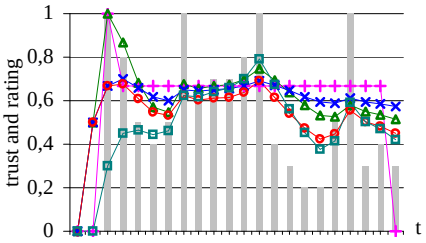

Fig. 7. SpecificRatingsNormal

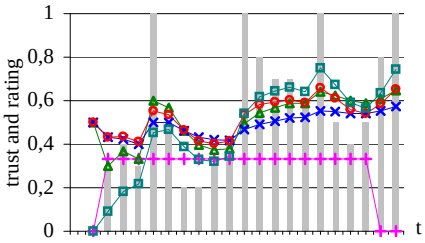

Fig. 8. SpecificRatingsReverse

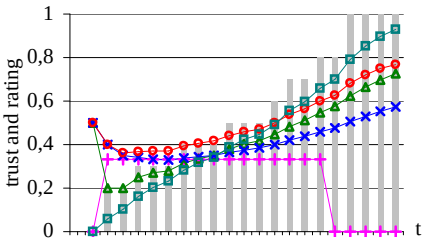

Fig. 9. SpecificOrderedAsc

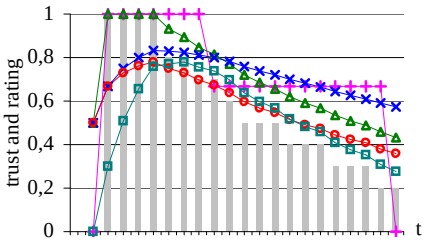

Fig. 10. SpecificOrderedDesc

to the sudden minimum ratings, while it also makes the reaction speed more independent of the total history size. It can be noticed that without an aging factor, the dynamics of trust gets more steady as the history of ratings grows. The reaction of the remaining algorithms to this test scenario can be seen in Fig. 12. Abdul-Rahman cannot really compete due to the lack of precision: After just one maximum or minimum rating trust flips from minimum to maximum and back to minimum trust value. ReGreT shows a similar deficiency as Beta Reputation without an aging factor: As more experiences are recorded, trust dynamics react slower to rating pattern changes. UniTEC shows quick reaction to the minimum ratings while maintaining this reaction independently of the history size.

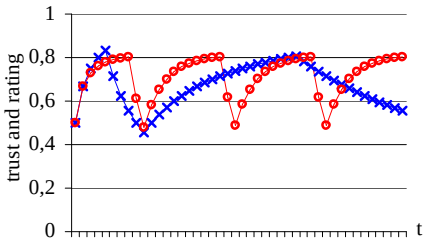

Fig. 11. KeepPositive and Beta Reputation

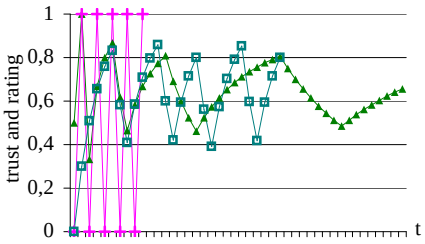

Fig. 12. KeepPositive and the other algorithms

8 Conclusion

In this work we investigated the various dimensions of trust relationships. Furthermore, we presented our approach towards a generic trust model which represents these dimensions and includes measures for trust, certainty, experiences and factors required for trust calculations. The model is based on observations gained through the analysis of a set of well-known trust models from the literature. It is generic in that it allows to plug in different specialized models and trust update algorithms and provides a bijective mapping between each local model and the generic trust representation. We discussed our adaptations of the original models which were necessary because we considered new trust relationship dimensions and ones that are not supported as such by all algorithms.

This generic trust model provides for the first time the possibility to compare various trust update algorithms through its common representation of algorithm inputs and outputs. We developed a set of test scenarios to assess the subjective quality of each supported algorithm. Our evaluation points out several important qualities but also deficiencies of the algorithms. To summarize our findings, we conclude that the Abdul-Rahman–Hailes algorithm in our generic trust model suffers from its discrete four step metrics in comparison to the field. The Beta Reputation system with an aging factor provides in our view the best overall results. The only drawback is the limitation of the trust value bandwidth which is proportional to the aging factor. The ReGreT algorithm provides responses to our test scenarios that meet our expectations, but its dynamics proved to be highly dependent on the history size: Too fast reactions without or with a small previous history of experiences, and slower dynamics as more experiences were collected. Finally the original UniTEC proposal provided a simple yet efficient algorithm and eased integration of the various dynamics. However, a deficiency of this algorithm lies in focusing merely on the current trust value and the latest experience and not taking into account patterns of past experiences.

Future work on trust update algorithms could consider giving more weight to negative experiences as opposed to positive ones. Furthermore, analyzing patterns of past experiences would be another interesting aspect to better detect misuse attempts and enhance the calculation of trust certainty. Besides improving existing trust update algorithms, we plan to investigate how to fit further algorithms into the generic model. Regarding the farther future, we consider to refine the representation of semantic distance in the model. In the current state of UniTEC, the semantic distances between the different trust context areas are specified by the applications and can be modified by each user. It would be challenging but surely interesting to investigate, whether and if yes how this process could be automated further.

References

1. Kinateder, M., Terdic, R., Rothermel, K.: Strong Pseudonymous Communication for Peer-to-Peer Reputation Systems. In: Proceedings of the ACM Symposium on Applied Computing 2005, Santa Fe, New Mexico, USA, ACM (2005)

2. Kinateder, M., Rothermel, K.: Architecture and Algorithms for a Distributed Reputation System. In Nixon, P., Terzis, S., eds.: Proceedings of the First International Conference on Trust Management. Volume 2692 of LNCS., Crete, Greece, Springer-Verlag (2003) 1–16
3. Gambetta, D.: Can We Trust Trust? In: Trust: Making and Breaking Cooperative Relations, Department of Sociology, University of Oxford (2000) 213–237
4. Jøsang, A., Gray, E., Kinateder, M.: Analysing Topologies of Transitive Trust. In Dimitrakos, T., Martinelli, F., eds.: Proceedings of the First International Workshop on Formal Aspects in Security & Trust (FAST 2003), Pisa, Italy (2003) 9–22
5. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, Maui Hawaii (2000)
6. Jonker, C.M., Treur, J.: Formal analysis of models for the dynamics of trust based on experiences. In Garijo, F.J., Boman, M., eds.: Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99). Volume 1647., Berlin, Germany, Springer-Verlag (1999) 221–231
7. Sabater, J.: Trust and Reputation for Agent Societies. PhD thesis, Institut d'Investigaci en Intelligncia Articial, Bellaterra (2003)
8. Jøsang, A., Ismail, R.: The Beta Reputation System. In: Proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia (2002)
9. Yu, B., Singh, M.P.: An evidential model of distributed reputation management. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna, Italy, ACM Press (2002) 294–301
10. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: Proceedings of the 17th IEEE Symposium on Security and Privacy, Oakland (1996) 164–173
11. Yu, B., Singh, M.P.: A Social Mechanism of Reputation Management in Electronic Communities. In Klusch, M., Kerschberg, L., eds.: Proceedings of the 4th International Workshop on Cooperative Information Agents. Volume 1860., Springer-Verlag (2000) 154–165
12. Mui, L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. PhD thesis, Massachusetts Institute of Technology (2003)
13. Marsh, S.P.: Formalising Trust as a Computational Concept. PhD thesis, Department of Mathematics and Computer Science, University of Stirling (1994)
14. Jøsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9** (2001) 279–311
15. Jøsang, A., Grandison, T.: Conditional Inference in Subjective Logic. In: Proceedings of the 6th International Conference on Information Fusion, Cairns (2003)