I. Agudo, C. Fernandez-Gago, and J. Lopez, "A Scale Based Trust Model for Multi-Context Environments", *Computers and Mathematics with Applications*, vol. 60, pp. 209-216, 2010.
http://doi.org/10.1016/j.camwa.2010.02.009
NICS Lab. Publications: https://www.nics.uma.es/publications

A Scale Based Trust Model for Multi-Context Environments

Isaac Agudo, Carmen Fernandez-Gago, Javier Lopez
NICS
www.nics.uma.es
University of Malaga,
Spain
{isaac,mcgago,jlm}@lcc.uma.es

December 19, 2011

Abstract

When interactions among users of a system have to take place, for example, over the internet, establishing trust relationships among these users becomes crucial. However, the way this trust is established depends to a certain extent on the context where the interactions take place. Most of the times trust is encoded as a numerical value that might not be very meaningful for a non very experienced user. In this paper we propose a model that takes into account the semantic and the computational sides of trust. This avoids users having to deal directly with the computational side, having instead to deal with meaningful labels such as *Bad* or *Good* in a given context.

Keyword 1 *Trust, trust scale, labels of trust, multi-context trust.*

1 Introduction

In some systems interactions among users happened among users that knew each other. This was the way interactions used to happen in most of the systems available in the past. This made trust easy to ensure most of the times.

Nowadays, due to the growth of online communities or internet transactions, it might not always be easy to determine whether to establish a trust relationship with a certain entity as we might not know it by first hand. Thus, establishing some mechanisms that support us to determine whether or not to trust an unknown entity becomes very useful. Moreover, we might have some trust information about an entity, due to past experiences, even in different contexts. Both second hand information and information related to different contexts should be considered when deciding whether to trust some entity in a given context.

Reputation systems are recently becoming a suitable tool for determining to whom to trust in certain environments where users have to interact among them in order, for example, to achieve a common goal or to access a certain resource. The way reputation is measured can vary (see [1] for a survey) ranging from assigning discrete values to using more complicated functions.

Trust is however a more subjective concept and it is often derived from reputation values. As well as reputation the way of measuring it can vary. Some attempts however have been made in order to find which metrics are more appropriate to use depending on the particular case [2] where they are going to be used. A usual way of measuring trust or reputation is by using discrete methods such as [3], where trustworthiness of an agent is classified as *Very Trustworthy*, *Trustworthy*, *Untrustworthy* and *Very Untrustworthy*. Fuzzy models are also a very suitable tool in order to measure trust or reputation. In these models, membership functions describe to what degree an entity can be described as trustworthy or not trustworthy [4, 5]. Other authors [6] present a framework for evaluating the resistance of trust metrics. The author used the Advogato system [7] for performing the tests.

However, users better understand a trust value such as *Bad* or *Good* rather than a less meaningful numerical value resulting from applying a method for deriving trust. Using this *Bad* or *Good* classification is a simple one. Different trust scales have been proven useful for the fields of Economics [8]. Moreover, it is sometimes hard to have information about different trust contexts that could help doing this classification. Thus, for example, the context in Epinions [9] is the ability to provide useful ratings about items.

There is always a trade-off between the strength and complexity of a trust and reputation assessment methodology and its usability. Sometimes the methodology to reason about trust becomes difficult to understand by a regular user, which makes then the methodology unpopular. A classic example is the Subjective logic [10] that despite of having a sound underlying

formalism has not been widely adopted by commercial applications. In fact, in actual applications such as eBay [11], trust (or more properly reputation) is measured as a matter of the ratio of good vs. bad interactions.

We present a model that based on the use of subjective *trust labels* such as *Trustworthy* or *Untrustworthy* that are meaningful for users, instead of using complex indexes that make the trust computation process obscure for them. We consider a trust scale for the way we classify these labels. They define trust levels among the different contexts of a system. By separating trust in the different contexts of a system we are able to combine them accordingly to user preferences, producing thus personalized trust assessments. For example, in the field of social networks, one context could be the user interactions with another users and another one could be the user contributions to the community. Our definition of context is very related to the concept of role.

The paper is organized as follows. Section 2 describes what we mean by a 'Trust Scale' and how we associate labels to it. In Section 3 we explain how the introduction of these scales is used in order to compute trust consensus that are useful for users in order to derive trust in different contexts. This section also shows several examples of different consensus functions. Section 4 concludes the paper and outlines the future work.

2 Trust Scales

Reputation systems provide users with some feedback about other users who they have to interact with. Most of the times reputation is given by a numerical value obtained after performing a continuous calculation of different factors. If a user is interested in gaining some information related to a possible interaction a value might not be that meaningful. What do these values mean for a user? Those values could be much more useful for a user if they are linked to a meaning such as 'bad', 'neutral' or 'good'.

A user, u, can be rated differently in different contexts such as, for example, sharing pictures or taking part in a forum. It is difficult to define disjoint contexts as there is no correlation between the trust measurements for a given user regarding each of the contexts. Contexts are rarely fully independent, that is why, we assume some overlapping between contexts and some correlations of the trust measurements.

Thus, the first step that we need to accomplish is the establishment of a reference set of contexts. Each context needs to be properly described

together with a set of trust labels that support assessing trust in this given context. Each label represents and is associated with a behavioural pattern that ultimately helps other users in the system understand clearly what to expect from this given user.

Let C be the set of contexts where a user u can be rated and m the cardinality of C, i.e., $\sharp C = m$. Let I_c be the set of labels indexes that are associated to C, i.e., $I_c = \{1, \ldots, n_c\}$.

Definition 1 (Trust Scale) *A trust scale for a given context $c \in C$ is composed of an ordered set $\mathcal{L}^c$ of trust labels L_i^c, where $i \in I_c$, that represent discrete trust meanings; and a trust evaluation that is an increasing function, $f : \mathcal{L}^c \to (0, 1]$, such that $f(L_{n_c}^c) = 1$.*
We denote $x_i^c := f(L_i^c)$ and $x_0^c := 0$.

Defining a trust scale requires some effort as it strongly depends on the context we are measuring. Each trust label must have a precise meaning within the definition context in such a way that users can easily understand the consequences of trusting a user rated with a particular trust label. This is what we could call the semantic side of a trust scale. However, trust scales also have an operational side.

In order to be able to do some computation with these trust labels we need to assign numerical values to them. We could do it in a standard way, by mapping the set of trust labels to a set of equidistant values in the interval $(0, 1]$. This, however could restrict the richness of our model. Then, apart from what we could call, a standard trust evaluation function, $f(L_i^c) = \frac{i}{n_c}$, we can define arbitrary increasing functions as trust evaluations.

A trust scale can be seen as a partition of the interval $(0, 1]$, where each label L_i is associated to an interval of the form $(x_{i-1}, x_i]$. Then, we can translate some of the terms applicable to partitions into trust scales. In particular, we can state that

Definition 2 *A trust scale $(\mathcal{L}^*, f^*)$is a refinement or sub-scale of the trust scale $(\mathcal{L}, f)$ if the set of labels of the second scale are a subset of the set of labels of the first scale, i.e. $\mathcal{L} \subset \mathcal{L}^*$; and the trust evaluations are consistent in the second scale, i.e. $f(L) = f^*(L)$ for all $L \in \mathcal{L}$.*

Figure 1 represents two trust scales $\mathcal{L} := \{L_1, L_2, L_3, L_4\}$ and $\mathcal{L}^* := \{L_1, L_{1-2}, L_2, L_{2-3}, L_3, L_4\}$, where $\mathcal{L}^*$ is a sub-scale of $\mathcal{L}$.

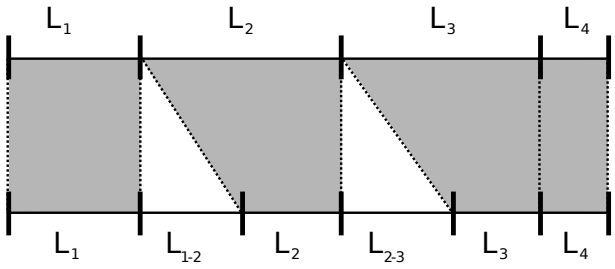


Figure 1: Trust Scales and Sub-scales

A logical way to refine an existing trust scale could be to use a *modifier* for the trust labels. For example, if there are only two trust labels $\{Bad, Good\}$ in the trust scale we can use the modifier *"less than"* in order to build a refined trust scale

$$\{less\ than\ Bad, Bad, less\ than\ Good, Good\}$$

In general, if we denote the modified labels by $\hat{L}_i^c$ and take as a base the original trust evaluation, we can extend it to an evaluation for the extended trust scale by defining $f(\hat{L}_i^c) = x_i^c - \alpha(x_i^c - x_{i-1}^c)$, for each $i \in I_c$ where $\alpha \in (0, 1)$. The standard extension mechanisms consist of setting $\alpha = \frac{1}{2}$.

3 Trust Computation

By using trust evaluations we link a label to a numerical value which will allow us to perform certain operations such as calculating a consensus trust value from a set of trust labels.

If we are interested in establishing this trust consensus value in a system this trust should be measured somehow. A simple way to measure trust could be established by using a binary discrete model where the trust values are set as *a lot of trust*, for a very trusted entity, or *very little trust* if the trust placed in the entity evaluated is very low. More complicated systems could use integer numbers (like Advogato's trust metric or FreeHaven [12]) or real numbers (like in [13, 14]).

After doing that, we have to translate this number back into a trust label. In order to do that, we have to make use of the *Inverse Evaluation Function*.

Definition 3 (Inverse Evaluation Function) *Given a Trust Evaluation function for a trust scale $\mathcal{L}^c$, $f : \mathcal{L}^c \to (0, 1]$, the Inverse Evaluation Function is defined as $f^{-1}(x) = L_i^c$ such as $x \in (x_{i-1}^c, x_i^c]$.*

We can then translate all the trust labels into actual numbers, compute the consensus and afterwards obtain a trust label that represents this consensus. The process is outlined in Equation 1.

$$\begin{array}{ccccc} & f & \circlearrowleft & f^{-1} & \\ \mathcal{L}^c & \longrightarrow & (0,1] & \longrightarrow & \mathcal{L}^c \end{array} \tag{1}$$

In [2] a model for trust metrics was introduced where the concept of 'Parallel Trust Function' is defined. These functions are used to combine several trust values into a single one that represents a consensus of them. This function is characterized for behaving as the identity function when there is only one value, and by being increasing in the sense that when one of the input trust values is increased the result is also increased.

Some functions that hold this property are:

- Minimum.
- Geometric Mean (GM). $\sqrt[n]{x_1 \cdot x_2 \cdots x_n}$
- Arithmetic Mean (AM). $\dfrac{x_1 + x_2 + \cdots + x_n}{n}$
- Maximum.

The following inequalities hold for any set of values $(x_1, \cdots x_n)$ in the domain of these functions,

$$Minimum \leq GM \leq AM \leq Maximum$$

On the one hand, the Minimum delivers a trust consensus that can be considered rather pessimistic. On the other hand, the Maximum will deliver a consensus that can be considered too optimistic (see Section 3.2 for examples of these facts). Then, the Geometric and Arithmetic Mean work better for the purpose of delivering a trust consensus. However, it is easy to find scenarios where each of them is the most suitable function.

3.1 Multi-Context Trust Computation

Although the use of trust labels when having only one context is interesting by itself, it is even more relevant when dealing with more than one context simultaneously. The interesting problem then is to integrate trust information from different contexts and extract a single trust value that somehow reflects the trust for all the considered contexts. As we can see in Diagram (2) there are several trust scales from each of the n different contexts, $\{c_1, \ldots, c_n\}$, and another scale where the unification value will be represented.

$$\begin{array}{ccccc} \mathcal{L}^{c_1} & & & & \\ & \searrow & & & \\ \vdots & \longrightarrow & (0,1] & \longrightarrow & \mathcal{L}^* \\ & \nearrow & & & \\ \mathcal{L}^{c_n} & & & & \end{array} \tag{2}$$

As we mentioned earlier trust scales are subjective and it is then difficult to define the trust scale $\mathcal{L}^*$ where the unification trust value is represented. The approach we follow is to choose a unique dominant context from the ones used as inputs. We will see in Section 3.2 with an example that choosing different dominant contexts delivers different consensus values. The trust output will reflect how the trust level in the dominant context is influenced by the performance of the user in the rest of the contexts involved.

In order to do this, we have to combine the consensus in each of the contexts by applying a kind of multi-context consensus function. The functions mentioned above are suitable for this purpose. An important issue that has to be also taken into account is the importance or relevance of the contexts with respect to the dominant context. That is the reason why we have also used the weighted mean as a candidate for the multi-context consensus function (see Section 3.2).

3.2 Example

Let us consider two exemplary contexts, a forum and a photo sharing site. Let us call these contexts F and P respectively.

Let us assume that the trust scale for context F consists of the following labels L_1, L_2 and L_3, and the results of the application of the trust evaluation for this trust scale is given by the values given below

- Bad $= L_1^F$ — $x_1^F = 0.3$
- Neutral $= L_2^F$ and — $x_2^F = 0.6$ and
- Good $= L_3^F$. — $x_3^F = 1$.

This means that any user in the forum can be rated as Bad, Neutral or Good by other members. Also, by looking at the trust evaluation we observe that the distribution of the labels is not homogeneous. The interval for Good is bigger than the interval for the other two labels.

The trust scale for context P consist of the following labels and trust evaluation values:

- Very Bad $= L_1^P$ — $x_1^P = 0.2$
- Bad $= L_2^P$ — $x_2^P = 0.4$
- Neutral $= L_3^P$ — $x_3^P = 0.6$
- Good $= L_4^P$ and — $x_4^P = 0.8$
- Very Good $= L_5^P$. — $x_5^P = 1$.

In this case there are five possible labels and the distribution of them is homogeneous.

Let now A, B, C, D and E be five users of both contexts with the following ratings with respect to them:

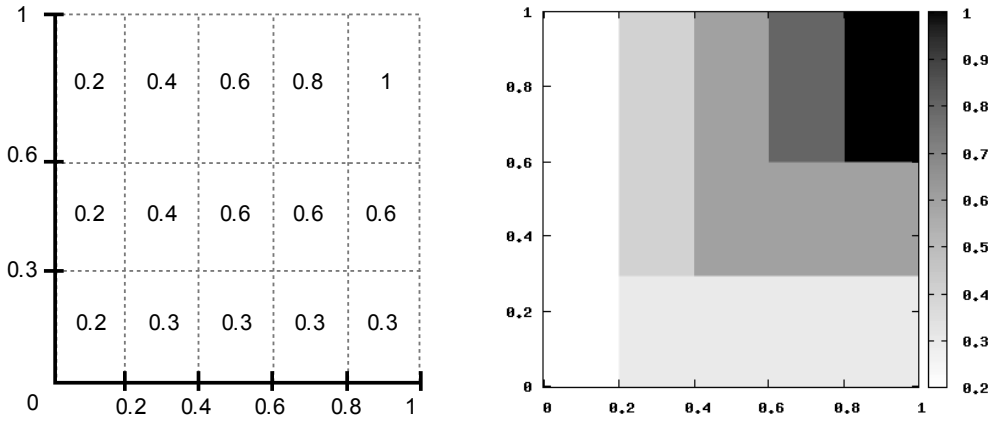
- (Good, Very Good)
- (Neutral, Neutral)
- (Bad,Very Bad)
- (Good, Very Bad) and
- (Bad, Very Good).

The first three users have behave in a similar way in both contexts whereas the last two behave in the opposite way as reflected in their ratings.

Now let us assume we are interested in computing a consensus for both contexts, using the scale of context P. This means we will use the rating in

context P as a basis and modify it according to the rating in the other context. For that process we can use different consensus functions, as mentioned above.

We first begin by analysing the consequences of using the Minimum function for computing the consensus. Figure 2(a) shows how the combined trust evaluation function works.



(a) Grid Representation (b) Density Map

Figure 2: Minimum

This grid contains in each quadrant the minimum of each evaluation of the labels. Given two labels, one for each context, the corresponding quadrant will represent the combined evaluation for this pair of context labels.

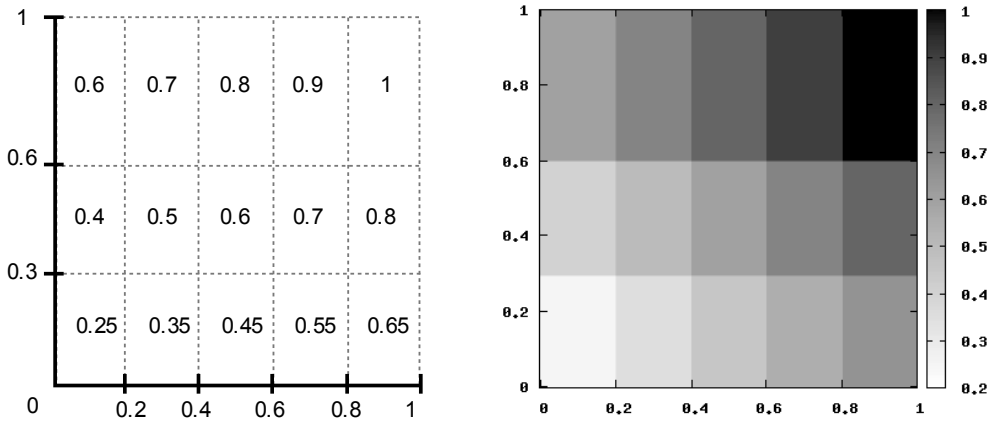
If we focus on a column we can see how the original trust label is degraded or improved depending on the evaluation of the context in the rows. Figure 2(b) shows an analogous representation where we use a grey scale, instead of the actual numbers, which helps understand the interrelations between the two contexts. This way is more intuitive to view that the use of the minimum degrades most of the times the original values.

In our particular example, the consensus trust label for A,B and C will be Good, Neutral and Very Bad respectively. For D, whose reputation is (Good,Very Bad) we obtain Very Bad and for E whose reputation is (Bad, Very Good) we obtain Bad. As a summary, the consensus values for the users are as follows

- A $\rightarrow$ Good

- B → Neutral
- C → Very Bad
- D → Very Bad
- E → Bad

We clearly observe that this is a pessimistic approach for merging trust evidences from different contexts.



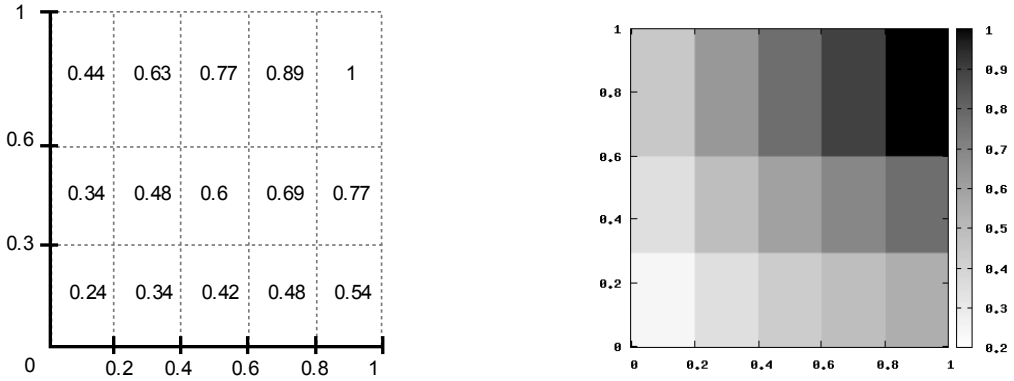
(a) Grid Representation (b) Density Map

Figure 3: Mean

Figure 3 shows the results of applying the arithmetic mean to the trust scales defined for contexts P and F. If we analyse how the consensus is affected when using the arithmetic mean we can observe that the values computed reflect a more realistic consensus. In our particular example, we obtain for A, B and C Very Good, Neutral and Bad respectively. Using the mean improves the consensus for C, which now turns Bad instead of Very Bad. If we look at D and E, a Neutral consensus is obtained. This clearly corresponds to the natural idea of the consensus. Summarizing the obtained consensus for the mean are as follows:

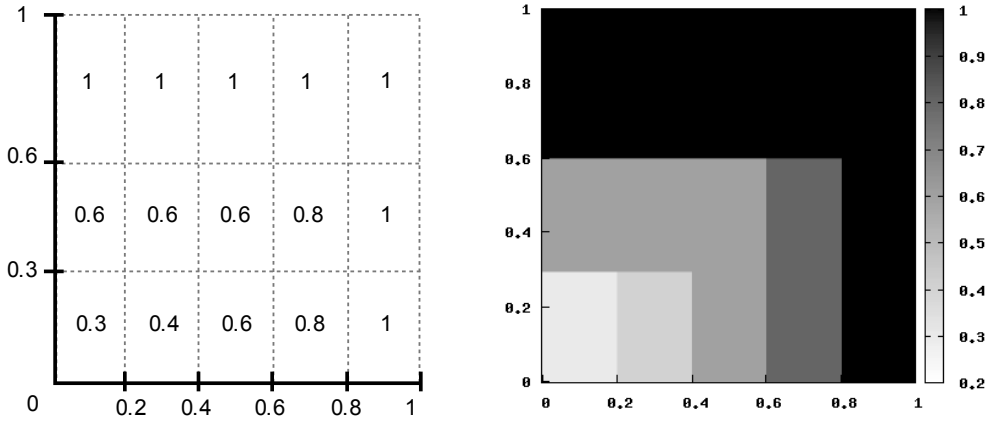
- A → Very Good
- B → Neutral

- C → Bad
- D → Neutral
- E → Neutral



(a) Grid Representation (b) Density Map

Figure 4: Geometric Mean



(a) Grid Representation (b) Density Map

Figure 5: Maximum

Figure 4 shows a graphical representation of the use of the geometric mean as a consensus function for the trust scales defined by F and P. We can

observe that the results are similar to the ones obtained by the arithmetic mean. Again, we can conclude that using a mean provides a more realistic approach for a consensus function.

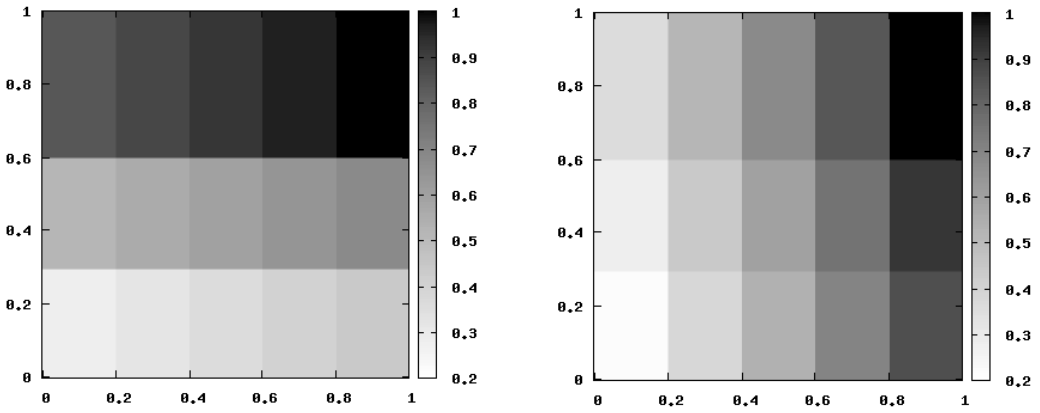
A more optimistic approach would be the use of the Maximum (See Figure 5). In this case the consensus inherits most of the better values from the contexts. A, B and C derive the same consensus than using the average mean but both D and E produce a Very Good consensus. Summarizing it

- A $\rightarrow$ Very Good
- B $\rightarrow$ Neutral
- C $\rightarrow$ Bad
- D $\rightarrow$ Very Good
- E $\rightarrow$ Very Good

Looking at these examples we realise the importance of choosing an appropriate consensus function. In most cases, the Mean offers a consensus that reflects the expected behaviour of a consensus function. There are some situations where the context that we are interested in combining have a different relevance. For example, if the aggregated trust computed from the two example contexts is required by a photo forum, we might require that the trust value for the photo sharing context is considered as dominant in the consensus computation. We then could use the Weighted Mean (see Figure 6).

In this case each context is associated to a weight that highlights its importance. The dominant context, i.e., the one in which we will translate the consensus back, is associated a higher weight than the other contexts. The higher the weight of the dominant context is the lower the influence of the rest of the contexts is. This is also reflected when we return to the labels of the trust scale of the dominant context. However, small differences in the numerical values of the consensus might vanish when using the inverse trust evaluation function.

Thus, in Figure 6(a) where context F is the dominant context and therefore is given a higher weight, 0.8 versus 0.2, we can observe that a variation in the marginal context will not influence much the consensus computation. We can observe that if we chain the three rows, one after the other, starting



(a) $0.8 \cdot F(y) + 0.2 \cdot P(x)$ (b) $0.2 \cdot F(y) + 0.8 \cdot P(x)$

Figure 6: Weighted Mean

from the one on the top, we obtain a decreasing grey scale. This means that the influence of the other context, P, is not strong enough to produce a relevant change in the dominant scale. Independently of the value in context P, the multi-context consensus will correspond approximately to the trust value in context F. In our example the obtained consensus are as follows:

- A $\rightarrow P^{-1}(1)$ =Very Good
- B $\rightarrow P^{-2}(0.6)$ =Neutral
- C $\rightarrow P^{-1}(0.28)$ =Bad
- D $\rightarrow P^{-1}(0.84)$ =Very Good and
- E $\rightarrow P^{-1}(0.44)$ =Neutral

The same applies to context P (see Figure 6(b)) when it is the dominant context with the same weights used as in the previous case, 0.2 versus 0.8. In this case we obtain the following results for the consensus:

- A $\rightarrow P^{-1}(1)$ =Very Good,
- B $\rightarrow P^{-1}(0.6)$ =Neutral
- C $\rightarrow P^{-1}(0.22)$ =Bad

- D $\rightarrow P^{-1}(0.36)$ =Bad and
- E $\rightarrow P^{-1}(0.86)$ =Very Good

4 Conclusions and Future Work

We have presented a trust model based on trust scales that takes into account the semantics of trust in different contexts. This model also allows us to do some basic computation, in particular, computing a trust consensus from the different contexts considered in the system. To the best of our knowledge no other trust management system have been developed taking into account the semantics of trust by using trust scales. The key aspect of our model is the use of trust scales that take into account both, semantic and computational aspects of trust.

The majority of trust management systems are developed without taking into account the semantics of trust. These systems are not very useful for users as the output is usually a numerical value that the user has to interpret without any guidance from the system. However, these systems need a strong computational model underneath the semantics of trust. Thus, we believe the model we presented could help existing on-line trust or reputation management systems, such as Venyo [15], to incorporate the semantics of trust from each of the context where they gather information from about their users. It could also be used to provide a personalized trust recommendation based on the user preferences.

In the future we intend to consider more complex computational models that could be linked to the concept of trust scale.

Acknowledgements

This work has been partially supported by the European Commission through the research project SPIKE (FP7-ICT-2007-1-217098) and the Spanish Ministry of Science and Education through the research projects CRISIS (TIN 2006-09242) and ARES (CSD 2007-00004)

References

[1] A. Jøsang, R. Ismail, C. Boyd, A Survey of Trust and Reputation Systems for Online Service Provision, Decision Support Systems 43 (2) (2007) 618–644. doi:http://dx.doi.org/10.1016/j.dss.2005.05.019.

[2] I. Agudo, C. Fernandez-Gago, J. Lopez, A Model for Trust Metrics Analysis, in: 5th International Conference, TrustBus 2008, (TrustBus'08), Vol. 5185 of LNCS, 2008, pp. 28–37.

[3] A. Abdul-Rahman, S. Hailes, Supporting Trust in Virtual Communities, in: Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.

[4] D. W. Manchala, Trust metrics, models and protocols for electronic commerce transactions, in: ICDCS '98: Proceedings of the The 18th International Conference on Distributed Computing Systems, IEEE Computer Society, Washington, DC, USA, 1998, p. 312.

[5] J. Sabater, C. Sierra, REGRET: A Reputation Model for Gregarious Societies, in: Fourth Workshop on Deception Fraud and Trust in Agent Societies, ACM Press, 2001.

[6] R. Levien, Attack Resistant Trust Metrics, Ph.D. thesis, UC Berkeley (2004).

[7] Advogato, http://advogato.org/.

[8] A. M. Evans, W. Revelle, Survey and Behavioral Measurements of Interpersonal Trust, Journal of Research in Personality 42 (6) (2008) 1585 – 1593. doi:DOI: 10.1016/j.jrp.2008.07.011.

[9] Epinions, http://www.epinions.com.

[10] A. Jøsang, R. Hayward, S. Pope, Trust network analysis with subjective logic, in: ACSC '06: Proceedings of the 29th Australasian Computer Science Conference, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2006, pp. 85–94.

[11] eBay, http://www.ebay.com.

[12] Freehaven, http://www.freehaven.net/.

[13] Openprivacy, http://www.openprivacy.org/.

[14] S. Marsh, Formalising Trust as a Computational Concept, Ph.D. thesis, Department of Computer Science and Mathematics, University of Stirling (1994).

[15] Venyo, http://www.venyo.org/.