How to talk about reputation using a common ontology: from definition to implementation

Isaac Pinyol
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
ipinyol@iiia.csic.es

Jordi Sabater-Mir
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
jsabater@iiia.csic.es

Guifré Cuní
Artificial Intelligence Research Institute
Campus UAB, Bellaterra
Barcelona, Spain
cuni@iiia.csic.es

ABSTRACT

The field of multiagent systems has experienced an important growth and evolution in the past few years. Due to the agent autonomy and their need for cooperation, special attention has been paid to reputation mechanism. Several reputation models have appeared in literature offering solutions to this problem. However, each one uses their own concepts, terminology and ways to represent evaluations that make impossible an hypothetical transmission of social evaluations between agents using different reputation systems. In this paper we describe and present the implementation of an ontology of reputation as well as an ontology mapping mechanism that can be used for these dialogs. When transmitting social evaluations, agents will map them into elements of this common ontology, that the receiver agents will translate to elements understandable by their particular reputation system.

1. INTRODUCTION

The field of multiagent systems has experienced an important growth and evolution in the past few years. These systems can be seen as virtual societies composed of autonomous agents where there is a need to interact with other members of the society to achieve their goals. As in human societies, these interactions are not solely restricted to direct *trades*, but also include simple exchanges of information[1]. A scenario like this arises intrinsically the problem of partners selection via the detection of good or bad potential partners, or how agents evaluate the credibility of received information. Internet has shown us at least one solution to that problem, one based on electronic certificates and encrypted digital signatures that make us *trust* the site we are visiting and increase the credibility of information that we gathered. Notice that this *metainformation*, these certificates, are generated by the same owners. However, human societies along its history have been using other mechanisms that take advantage of exchanging information about the goodness of other members' performances. They are the trust and reputation mechanisms, a very powerful social control artifacts that have been studied from different perspectives, such as psychology (Bromley [3], Karlins et al. [15]), sociology (Buskens [4]), philosophy (Plato [18], Hume [14]) and economy (Marimon et al. [16], Celentani et al. [8]).

As described in Conte and Paolucci [9], reputation-based systems can be seen as a spontaneous and implicit norm-based system for social control. Every society has its own rules and norms that participants should follow to achieve a *well-fare* society. Social control mechanisms should be able to exclude not normative participants, but not even designed institutions can do that because they would need to keep a control over each transaction, and it is totally out of the question in distributed environments populated by autonomous entities. Here is where reputation-based systems can help to solve the problem. The social control they generate emerge implicitly in society, since non normative behaviors would tend to generate bad reputation that agents will take into account when selecting their partners, and therefore it may cause exclusion of the society due to social rejection. The idea of not having a totally explicit social control is crucial in multiagent systems and the last decade the interest on these mechanisms has considerably increased in the field. As a consequence numerous reputation models have appeared in the literature.

E-Commerce sites already use some of them (eBay [10], Amazon [2], OnSale [17]). These models consider reputation as a centralized global property of each particular agent (in these cases, sellers and buyers) calculated from the ratings that the system has received from users. These *reputation* values may be taken into account by potential buyers while selecting sellers. More sophisticated models ([1], [13], [21], [26], [5], [22], [19]) consider reputation as a subjective and contextualized property. Therefore every agent has its own reputation system that provides evaluations of other agents calculated from external communication and direct experience, giving the agent its own vision of the society. Furthermore, other models (see [6], [20]) take into account social information when providing these evaluations.

Once the actual situation is set, the beauty of such diversity clashes against an hypothetical virtual society whose agents do not use the same model. Since all of them use their own nomenclature, representation of the evaluations and even ways to interact, there is no way to establish communication between two agents using different reputation models that exchanges information about social evaluations. If the

[1]We understand as direct trade an interaction between agents (typically two) that involves cooperation or offering some service, and where it is possible to define a previous contract with the expectations that have the different parts. This is different from exchanging information, since in that case, there is no contract to be accomplished

source knew which model is using the recipient it might *convert* his/her own representation to the other one, but there is no reason to think that agents will know the internal functionalities of other participants.

This work gives an implementable solution to this problem. We propose a common ontology for reputation that allows the communication of social evaluations among agents that are using different reputation models. The implementation is based on a common API[2] that will work as a middleware between the common ontology and each particular model. An special emphasis is put on how to deal with the differences on the representation of the social evaluation values.

In Section 2 we explain the related work in reputation ontologies that exists in the literature. In Section 3, we describe in detail the proposed ontology and its main elements. Then, in Section 4 we define how we deal with the problem of value representation in social evaluations. In Section 5, we explain the design of the API interface. Finally, in Section 6, we show how the ontology can be mapped to several representative reputation models.

2. RELATED WORK

There is not much work done on this specific topic. When talking about ontologies for reputation there are in the literature two main works.

On one hand, in Casare et al. [7] is proposed a functional ontology the goal of which is to put together in a very conceptual level all the knowledge about reputation. It is based on the concepts defined in the Functional Ontology of Law [24]. This approach is interesting from a theoretical point of view because offers an structured definition of reputation and its related concepts, including processes of transmission, that could be used, as they claim in [7], as a *meta-model* of reputation concepts that could be mapped into existent models. In fact, in [25] is proposed a basic agent architecture to allow interoperability between agents using different reputation models that incorporates their functional ontology in this way. However, it is still a very conceptual approach that does not give solutions for the problems that in a possible real implementation we could find, like representation types for the evaluations, ontology mapping mechanisms etc...

On the other hand, as part of the European project eRep [12] a set of terms about reputation concepts has been defined. They can be seen in the *wiki* of the project [11]. The aim of this effort is to define an ontology that all partners participating in the project could use as a consensual starting point. This ontology describes in detail all the elements participating in social evaluations, as well as the processes of transmitting them. It also defines the main decisions concerning reputation that agents may take. This ontology is based on the cognitive theory of reputation defined in the book by Conte and Paolucci [9] and the Repage model [19].

The ontology presented in this article is a subset of the one used in eRep. Basically we are taking the elements of the eRep ontology concerning agents' beliefs that deal with social evaluations (see Section 3) adapting them to allow a

[2]Standard acronym for Application Programming Interface

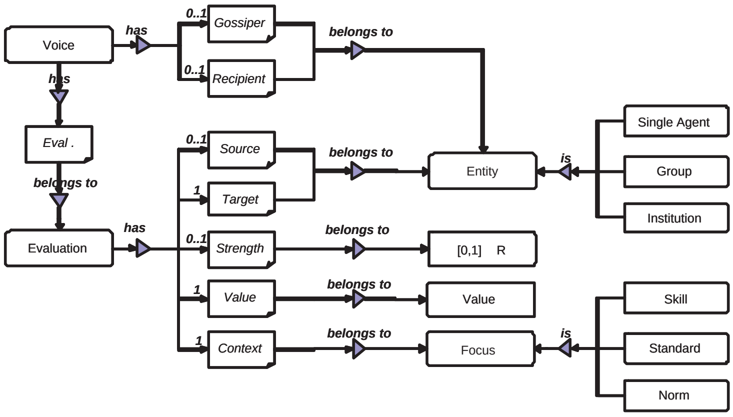

Figure 1: The main classes and components of a social evaluation and voice

direct and easy implementation. The main difference between the resulting ontology and the ontologies in eRep [12] and Casare et al. [7] is the focus on the implementation with an special emphasis on the representation of the social evaluation values (see Section 4), something not present in the previous approaches. For a description of the rationale behind each element of the ontology we refer the reader to the article about the Repage model [19].

3. THE ONTOLOGY

3.1 Preliminaries

In order to explain the elements of the ontology it is necessary to get in touch with some concepts defined in [9]. This cognitive theory keeps an essential difference between *image* and *reputation*. Both are social evaluations, evaluations that concern other participants in the society, individuals like single agents, or supra individuals like groups or collectives, but while *image* refers to evaluations that agents take as certainty, *reputation* refers to other's evaluations and therefore is considered a meta-belief, that is, a belief about other's belief. This brings us some important consequences, since accepting a meta-belief does not imply to accept the nested belief. Assuming that target agent A has some given *reputation* means that it is reputed with more or less goodness, and that such evaluation circulates on the society, but not necessary implies to share the evaluation itself. So, *reputation* refers to what is said, not what is true.

In the following subsections 3.2 and 3.3 we explain the elements of this ontology, from the characterization of a social evaluation to a taxonomy of what we understand as evaluative belief, that is, agent's beliefs that include some social evaluation. For a more formal definition of the ontology we refer to the work performed in the eRep project [12]. Here the objective is to provide a less formal description of the elements.

3.2 The social evaluation and voice

In Figure 1 we show the main elements involved in the definition of a social `Evaluation` and `Voice`. In the following paragraphs we describe each of them.

3.2.1 Entity

An Entity is any element of the society susceptible of being evaluated or having an active part in the generation or diffusion of evaluations. From the point of view of our theory an Entity can participate in the process of reputation in four different ways. On one hand, an Entity being evaluated is playing the social role of Target, meanwhile the one generating it, the role of Source. On the other hand, an Entity that spreads an Evaluation plays the role of Gossiper, and the one receiving it the role of Recipient.

3.2.2 Focus

The class Focus is the context of the evaluation. The goodness or badness of some entity's Evaluation is towards to an specific Norm, Standard or Skill. Agents can evaluate the same Target agent from different points of view. For instance, we can have a bad *image* of Agent A as a chess player, but a very good *image* of the same agent as a soccer player.

3.2.3 Value and Strength

The class Value contains the goodness or badness of the Evaluation, it is how good or bad is the Target entity in the context defined by the element Focus by the Source entity. In Section 4 we describe how we have represented this value. The Strength, represented with a bounded real belonging to the interval $[0, 1]$ is a subjective measure set by the Source that indicates how reliable is the evaluation, being 1 the maximum reliability. For instance, agent A may have had only one direct interaction with agent B, getting very good results. Then, agent A may generate a very good evaluation of this agent, but because he/she had only one interaction, agent A may not be totally sure of this Evaluation, specially when is communicating this result to another agent. The Strength is simply a measure that agents may use in order to be more accurate in their Evaluations.

As we will see, the Strength value is closely related to the uncertainty conversion (CU) described in Section 4.3. Both refer to some uncertainty associated to the Value class, but meanwhile the Strength is a subjective value that agents deliberately set, CU is intrinsic to how the Value is represented and the history of performed conversions, as will be shown in Section 4.3. It is a decision of each agent to use and combine these indicators or simply ignore them.

3.2.4 Evaluation and Voice

Finally, the class Evaluation encapsulates all the elements that participate in a social evaluation. It includes two instances of the class Entity playing the role of Source and Target, the Context, that belongs to the class Focus, and finally the Value of the evaluation. In the literature we found reputation and trust models that represent the Value with a simple boolean (good, bad), with a bounded real or even with fuzzy sets or probability distributions. The choice of this representation is one of the most delicate issues when developing a common ontology that should take into account a whole variety of representations, or at least, the most common ones. We discuss this problem and the solution we propose in Section 4.

Once defined the class Evaluation, we introduce the class Voice, that includes the necessary elements to represent the spreading of an Evaluation. A Voice is defined as a 'report on reputation'. For instance, "It IS SAID that John is good at playing soccer" is an example of a Voice. Apart from the Evaluation itself, it has two attributes belonging to Entity that identify the Gossiper and the Recipient of the Voice.

3.3 Evaluative Belief

Not all the Evaluations and Voices that agents have in their belief set have an specific meaning. I may have my own Evaluation of somebody, but I may have as well an Evaluation that somebody else gave me. Both are Evaluations, but their semantics are completely different. Taking into account this, the ontology describes a taxonomy of evaluative beliefs, beliefs that contain some Evaluation. Figure 2 shows the graphical representation of the taxonomy and the attributes that the leaf classes have. As a root we find the class EvalBelief, representing social evaluations that agents may have in their belief set. As showed in Figure 2 we divide the social evaluations in two categories, the classes SimpleBelief, a belief that the holding agent acknowledge as true, and MetaBelief, a belief about others' belief (in other words, a representation of other's mind). Let's describe each one of the bottom classes.

3.3.1 Image

An agent holding it, believes in the Evaluation that contains the class as attribute. In other words, an Image contains the believed opinion of the agent about a given Target with respect to a given Focus. The important point here is that the agent believes that this Evaluation is true.

3.3.2 Direct Experience

This class refers to the Evaluation that an agent creates from a single interaction or experience with another Entity. After an interaction, the generated outcome (the objective result of the transaction) is subjectively evaluated by the agent. This evaluation depends on the current mental state of the agent. The Evaluation is associated with a particular transaction, represented in the class by the attribute IdTrans[3] .

3.3.3 Shared Voice

An agent holding a Shared Voice will have the certainty that a perfectly identified set of Entities will acknowledge the existence of the Voice included in the class. Following the previous example, the fact that agent A, B, C and D inform agent X that "It IS SAID that John is good at playing soccer" is understood as a Shared Voice, since a set of agents (A,B,C,D) share the same Voice about "John". This class only refers to what the set of agents have informed, it is not a representation of what they believe. Therefore, a Shared Voice cannot be considered a meta-belief.

3.3.4 Shared Image

In this case, an agent holding a Shared Image believes that a perfectly identified set of Entities have as a belief the Evaluation included in the object. Clearly, this concept is considered a meta-belief in the sense that is a belief of other's belief, even though the set of Entities is known.

[3]In Figure 2 this attribute is presented belonging to the class Real. It simply means that IdTrans $\in \mathbb{R}$.

3.3.5 Reputation

In our approach, `Reputation` is a generalization and loss of reference of the `Shared Voice`. An agent holding a `Reputation` believes that most of the entities would acknowledge the existence of the `Voice` included in the class. It refers to what a target agent "IS SAID to BE" by most of the population or group. From the point of view of the holder agent, it is understood as a belief of others' belief in the sense that the holder agent believes that most of the population believe certain evaluation. For instance, taking again our example, to acknowledge that most of the people say that "John is good at playing soccer", can be understood as to believe that most of the people believe that "John is good as a soccer player", but this does not imply to really believe that John is good at it. Again, we consider `Reputation` as a meta-belief.

4. VALUE REPRESENTATION AND CONVERSIONS

One of the most important aspects of the reputation models are the value representations and semantics they used for the evaluations. In literature we find numerous models each of them using a different way to represent the value of an evaluation, from a simple boolean value indicating good or bad, to probability distributions and fuzzy sets. For instance, the eBay model uses a system of colored stars to show the reputation of a seller that could be seen as a simple real number between 0 and 100.000, meanwhile the Repage model uses a probability distribution over the discrete set Very Bad, Bad, Neutral, Good, Very Good. . When designing a common ontology on reputation it is crucial to have a consensual representation of the evaluations, since they are the key for the wished understandability between agents using different reputation models.

To decide a representation that everybody has to use is not an easy issue. For instance, if we took a simple boolean representation, models using a real number for evaluations would loose a lot on information when using the ontology. For this reason and after checking the most popular reputation models we decided to allow different types of representation (Boolean, Real, Discrete Set and Probability Distribution) providing automatic transformation functions between types, to allow connectivity between agents using different models that use different representations. The representation type is encapsulated in the class *Value* of the ontology.

In this section we explained these four types in detail and its semantics, as well as the transformation functions we designed.

4.1 Type definitions

- **Boolean Representation** (BO): In this case, evaluations take two possible values, good or bad. We define true as Good, and false as Bad.

- **Bounded Real Representation** (RE): Here, the value is a real number included in the bounded interval $[0, 1]$ where 0 is the worst evaluations, 1 the best evaluation and 0.5 the absolute neutral evaluation. The curve we have chosen indicating the level of goodness/badness is completely linear, from 0 to 1.

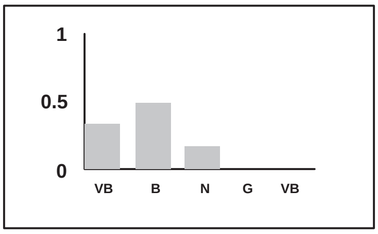

Figure 3: The graphical representation of the prob. distribution of (0.3,0.5,0.2,0,0)

- **Discrete Sets Representation** (DS): In this case, the value belongs to the following sorted discrete set {*Very Bad, Bad, Neutral, Good, Very Good*} ({VB,B,N,G,VG} from now on). Its semantics is intrinsic on the definition of each element of the sorted set.

- **Probabilistic Distribution Representation** (PD): Finally, this last representation applies a probability distribution (PD) over the sorted discrete set seen in the DS representation.

  Let L be the vector $[VB, B, N, G, VG]$ where $L_1 = VB, L_2 = B$ and so on. If X is a probability distribution over L then we define X_i as the probability of being evaluated as L_i. Given X is a probability distribution we have that $\sum_{i=1..5} X_i = 1$. For instance, we could have the distribution $(0.3, 0.5, 0.2, 0, 0)$ meaning that with probability of 0.3 the agent evaluated is *Very Bad*, with 0.5 that is *Bad* and with 0.2 that is *Neutral*. Graphically it can be represented as shown in figure 3

4.2 Transformation Functions

As we stated, we offer automatic transformations between the types we explained in the previous subsection. Some of them require justified arbitrary decisions. Table 1 summarizes all these transformations, however, in the following subsection we explain in detail each of them.

4.2.1 Transformations From PD

This is the most expressive type,being the only one offering probabilities. Because of that, it is difficult to find a direct transformation to the rest of types. Let's check each possible transformation:

$\rightarrow$ *To Boolean (BO)*

In the BO we only have two values, false/true. The idea is that a PB value will converge to a *Good* if the probability distribution *tends* to values $\{Good, VeryGood\}$, and *Bad* if it *tends* to the values $\{VeryBad, Bad\}$. In our context, the word *tend* implies that we need an operation capable to transform a probability distribution element to an unidimensional number with some range, in where a threshold could tell us whether to transform the PD value to true or false. This operation is what we have called the center of mass (CM) of a PD element, $CM : PD \rightarrow [0, 1] \in I\!R$. It returns a bounded real number $\in [0, 1]$ indicating in terms of average how *GOOD* (converging to 1) or *BAD* (converging to 0) is the evaluation value represented in a probabilistic distribution. Of course 0.5 would be the absolute neutral.

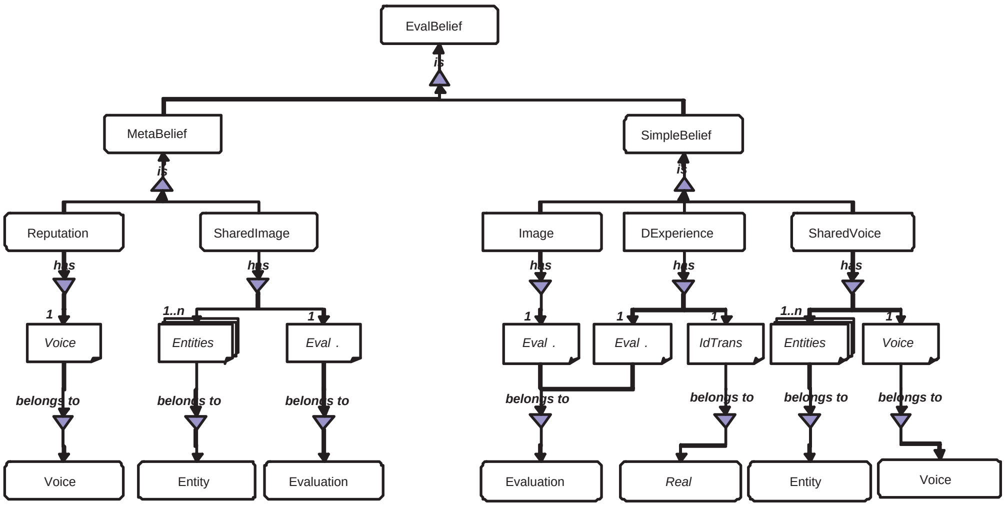

Figure 2: The taxonomy, membership relations and components of evaluative beliefs

Then, it is easy to think that values over 0.5 would indicate mostly good, and below mostly BAD. The value 0.5 will be our threshold. Let $X \in PB$, the function CM is defined as follows:

$$CM(X) = \frac{1}{10}\sum_{i=1}^{5}(2i-1)X_i \qquad (1)$$

Then, to transform a given PB value X to a boolean is enough to evaluate the following boolean expression[4]: $CM(X) >= 0.5$

→ To Real (RE)

Once defined the center of mass function, let $X \in PB$ the transformation to a RE is: $CM(X)$

→ To Discrete Set (DS)

Notice that due to the semantics of the bounded real type (RE), the interval $[0,1]$ could be mapped into the ordered discrete set type (DS) $\{VB, B, N, G, VG\}$ in an easy way, keeping the semantics in the transformation. Let's define the function $R : [0,1] \in I\!R \rightarrow \{VB, B, N, G, VB\}$ that do this mapping as follows: Let $X \in RE$, then

$$R(X) = \begin{cases} VB & \text{if } 0 \leq X \leq 0.2; \\ B & \text{if } 0.2 < X \leq 0.4 \ ; \\ N & \text{if } 0.4 < X \leq 0.6 \ ; \\ G & \text{if } 0.6 < X \leq 0.8 \ ; \\ VG & \text{if } 0.8 < X \leq 1 \ . \end{cases} \qquad (2)$$

Then, we have already seen how to transform an element in type PD to type RE. Then we can apply the R function over the result element in type RE, obtaining an element of type DS. Given that, the full transformation of an element $X \in PD$ would be calculated by the expression $R(CM(X))$.

[4]The decision of including 0.5 as a good evaluation is totally arbitrary, but consistent in all the transformations

4.2.2 Transformations From DS

→ To Boolean (BO)

In this case, the semantics of $TRUE$ in a boolean representation implies a possible condition of G or VG in a discrete set representation, and the $FALSE$ to a VB or B. Following the same decision we made in the previous subsection, the neutral value N should be considered $TRUE$ as well. Therefore, the transformation is complete. In order to do it more formal, we define the function $S : \{VB, B, N, G, VB\} \rightarrow [1,5] \in I\!N$ that returns the index position of a given element in the sorted set $\{VB, B, N, G, VG\}$, and it is defined as follows:

Let $X \in DS$, then

$$S(X) = \begin{cases} 1 & \text{if } X = VB; \\ 2 & \text{if } X = B \ ; \\ 3 & \text{if } X = N \ ; \\ 4 & \text{if } X = G \ ; \\ 5 & \text{if } X = VG \ . \end{cases} \qquad (3)$$

Then, the transformation to BO is calculated with the expression $S(X) \geq 3$.

→ To Real (RE)

For this transformation we need to realize that function R (equation 2) divides the interval $[0,1]$ into five parts, each of them assigned to one of the values of the type DS. For instance, all elements between 0.2 and 0.4 are mapped in the element B (bad) of DS. Then, given an element of type DS, the possible real value equivalent should be included in the interval defined in function R. Then, for instance, a VB value as real would be in the interval $(0.2, 0.4]$.In fact, whatever value in the interval would be fine, however, we decided to pick the one just in the middle,0.3 that would reach the less error if we pick randomly one number from the interval. To formalize the transformation we decided to use a function that gives this central point.We define the function $C : [1,5] \in I\!N \rightarrow \{0.1, 0.3, 0.5, 0.7, 0.9\}$.Let $X \in [1,5]$ the function C is defined as $C(X) = \frac{2X-1}{10}$. Having it, we can describe the transformation.

Let $X \in DS$, its transformation to RE would we calculated with the expression $C(S(X))$.

$\rightarrow$ *To Probabilistic Distribution (PD)*

This case is quite simple, since a DS can be seen as a particular case of a PD, assigning the probability of 1 to the corresponding element of the set. For this reason we define the function $B : [1,5] \in I\!N \rightarrow PD$, that creates a PD element assigning a probability of 1 to the corresponding element and zero to the rest. Let $i \in [1,5] \in I\!N$ the function B is defined as:

$$B(i) = \{X \in PD : \forall_{r \neq i} X_r = 0 \& X_i = 1\} \tag{4}$$

Then, let $X \in DS$, its transformation to PD is calculated with the expression $B(S(X))$. Notice that the possible uncertainty that can have the source DS value, will be reflected in the strength and therefore will be transmitted as such to the strength of the target PD value. Therefore, it is not necessary to consider this uncertainty in the transformation of the value.

4.2.3 Transformations From RE

$\rightarrow$ *To Boolean (BO)*

Following the same reasoning used in subsection 4.2.1, let $X \in RE$, the transformation between a RE type to a BO type is calculated evaluating the expression $X \geq 0.5$.

$\rightarrow$ *To Discrete Set (DS)*

Let $X \in RE$ and having defined the function R (eq 2), the transformation to a DS is calculated using the expression $R(X)$.

$\rightarrow$ *To Probabilistic Distribution (PD)*

The idea for converting an element $X \in RE$ to a PD type is to generate a PD element which center of mass is equal to X. It is obvious that there are infinite possible combinations. We decided to pick only two contiguous elements of the PD set and assign the corresponding probabilities in order to achieve the desirable center of mass.

Let's consider that i_1 and i_2 are the two index positions of the elements of PD that we choose to create the PD value. We decided the elements would be contiguous. To calculate them we use the function $R' : [0,1] \in I\!R \rightarrow [1,5] \in I\!N$ defined as

$$R'(X) = \begin{cases} 1 & \text{if } 0 \leq X \leq 0.2; \\ 2 & \text{if } 0.2 < X \leq 0.4 \text{ ;} \\ 3 & \text{if } 0.4 < X \leq 0.6 \text{ ;} \\ 4 & \text{if } 0.6 < X \leq 0.8 \text{ ;} \\ 5 & \text{if } 0.8 < X \leq 1 \text{ .} \end{cases} \tag{5}$$

This equation simply indicates to which interval belongs certain RE element[5]. We find i_1 as follow:

$$i_1 = min\{R'(X), R'(max\{X - 0.1, 0\})\} \tag{6}$$

[5]Notice that the equality $R'(X) = S(R(X))$ is hold

The index i_2 is the next following number, taking into account that the maximum allowed number is 5

$$i_2 = min\{5, i_1 + 1\} \tag{7}$$

Let $X \in RE$, then we need to find two probabilities, z and y, such that $z + y = 1$ and its center of mass (considering a PD element) is the original X, it means, that the following equation holds:

$$X = C(i_1)z + C(i_2)y \tag{8}$$

Solving the equation we have that $z = 1 - y$ and $y = \frac{X - C(i_1)}{C(i_2) - C(i_1)}$. Having this, we use the function $B' : [1,4] \in I\!N x[0,1] \in I\!R \rightarrow PD$ defined as

$$B'(i,p) = \{X \in PD : \forall_{r \neq i,i+1} X_r = 0 \& X_i = p \& X_{i+1} = 1 - p\} \tag{9}$$

For instance, $B'(3, 0.3)$ returns as a PD element [0,0,0.3,0.7,0], $B'(1, 0.8)$ returns [0.8,0.2,0,0,0]. Finally, we have all the element to calculate the transformation from a given $X \in RE$ to a PD:

$$B'(i_1, 1 - \frac{X - C(i_1)}{C(i_2) - C(i_1)}) \tag{10}$$

4.2.4 Transformations From BO

In this case we start our explanation from the most expressive type (PD) .

$\rightarrow$ *To Probabilistic Distribution (PD)*

In this point, it is important to notice that knowing one of the two possible values of the boolean representation, implies certain conditions that help to decide the transformations. Having only a $FALSE$ we can say that whatever representation to be transformed, the transformed value should be in the side of the bad evaluations. In fact, with the type of probabilistic distribution, because it works with probabilities, we could think that in the $FALSE$ value for instance, there is the same probability of being Very Bad, and Bad, and less probability although some , of being Neutral. The same happens with the value TRUE. Then, we define the following constants $B_F = [2/5, 2/5, 1/5, 0, 0]$ and $B_T = [0, 0, 1/5, 2/5, 2/5]$ belonging to PD. The transformation function from a BO to PD is then quite simple. Let $X \in BO$, the conversion is:

$$\begin{array}{c} B_T : X \\ B_F : \neg X \end{array} \tag{11}$$

$\rightarrow$ *To Discrete Set (DS)*

Here we should decide which values of the considered bad evaluations or good evaluations correspond to the $FALSE$ and $TRUE$ values respectively. However, once fixed the constants B_F and B_T to represent FALSE and TRUE values in probabilistic distribution, we have to take them as a base to decide the transformation. The idea is that if B_F represents a $FALSE$E, its center of mass (function CM) has to indicate the position in the interval $[0,1]$ that the FALSE value represents, and having it, the function R would determine which element of the discrete set represents the $FALSE$ value. The same reasoning can be made for the $TRUE$

value. Then, let $X \in BO$, the transformation to DS is:

$$\begin{array}{l} R(CM(B_T)) : X \\ R(CM(B_F)) : \neg X \end{array} \quad (12)$$

Actually, the $FALSE$ value goes to B, and $TRUE$ goes to G.

→ *To Real (RE)*

If we have used the function CM in the previous transformation, it is clear that to keep consistency, having $X \in BO$ the transformation should be:

$$\begin{array}{l} CM(B_T) : X \\ CM(B_F) : \neg X \end{array} \quad (13)$$

4.3 Conversion Uncertainty

As we stated, in a society where participants may be using different kinds of reputation and trust models, the necessity of exchanging social evaluations to achieve their goals may drive in a situation where an agent that uses a boolean representation needs to communicate with one that uses probabilistic distribution, and then, a conversion of representations must take place. However, type conversions carry lose of precision and addition of uncertainty. As an example, some evaluation represented as a boolean that is Bad, when is converted to a real representation may have an evaluation from 0 to 0.5 (not included), when is converted to discrete set, it may be one of these elements $\{VB, B\}$ etc... This factor of uncertainty that is added when we convert a value to a more expressive representation is what we call Conversion Uncertainty (CU), and is an information that the recipient should know.

4.3.1 Entropy of the representations

In order to calculate the CU we decided to use the information theory approach introduced by Shannon [23]. In this context, the entropy of a random variable X ($H(X)$) can be understood as the *uncertainty* of X, and is defined as

$$H(X) = -\sum_{x \in X} p(X = x) \log(p(X = x)) \quad (14)$$

From Shannons's theory we can define the conditional entropy as follows:

$$H(X|Y = x) = -\sum_{x \in X} p(X = x|y = Y) \log(p(X = x|Y = y)) \quad (15)$$

and finally,

$$H(X|Y) = -\sum_{y \in Y} p(Y = y) H(X|Y = y) \quad (16)$$

Now, we consider each one of the representations as discrete random variables. Without lose of generality we can discretize the Real representation using two digits (in base 10), having a hundred possible values. The fact of using 100 divisions for the interval and not a bigger amount is because we think that a greater precision is completely unnecessary (and even counterproductive) given the nature of the measure that is represented with this value, that is, a measure of a social evaluation. At the same time, taking into account the hundred possible values of a Real number, we can count the number of elements of the Probabilistic Distribution representation considering all possible combinations of distribution values that need to achieve the unit[6].Let A be this number, the following equation is hold:

$$A = \sum_{i=0}^{4} \binom{5}{i} \binom{100}{4-i} = 4780230 \quad (17)$$

Each random variable has as elements each possible element of the representations and it's probability distribution is totally equiprobable. Then, we define the conversion uncertainty of the source random variable X to the target random variable Y as

$$CU(X, Y) = H(Y|X) \quad (18)$$

In other words, CU is the increment of uncertainty produced when a value is represented in X and is converted to a value of type Y that is more expressive. There is a set of candidate values that makes conditional entropy increase. The values of the entropy of each type is showed in table 2. See appendix A for the details of the calculus.

The CU values for each conversion is showed in table 3. Each row is the source and each column is the target.

Type	Entropy
BO	1.00
DS	2.31
RE	6.64
PD	22.19

Table 2: Entropies of the type representation

	BO	DS	RE	PD
BO	0	1.29	5.64	21.19
DS	0	0	4.32	19.89
RE	0	0	0	15.55
PD	0	0	0	0

Table 3: CU values

4.3.2 CU usage. An example

An example will illustrate the usage of the CU value. Let's suppose agent A is using a Boolean representation, and generates and sends an evaluation to agent B that uses a discrete set representation. Agent B would reach the evaluation with a CU value of 1.29. If agent B send the same evaluation to agent C that uses a probabilistic distribution, agent C would receive the evaluation with a CU value of 1.29 (the base value coming from the communication) plus 19.89 (from the type conversion between DS to PD), it means, a CU value of 21.18. The idea is that the uncertainty of the evaluations is accumulative, without allowing loops (if the evaluation goes back to an agent using a representation type that have already been used in some transformation there is no addition of uncertainty)

[6]This is a combinatorial problem related to the famous Balls and Bins problem

	BO	RE	DS	PD
X:BO	X	$CM(B_T) : X$ $CM(B_F) : \neg X$	$R(CM(B_T)) : X$ $R(CM(B_F)) : \neg X$	$B_T : X$ $B_F : \neg X$
$X : RE$	$X \geq 0.5$	X	R(X)	$i_1 = min\{R'(X)$ $R'(X - 0.1)\}$ $i_2 = min\{5, i_1 + 1\}$ $B'(i_1, 1 - \frac{X - C(i_1)}{C(i_2) - C(i_1)})$
X:DS	$S(X) \geq 3$	$C(S(X))$	X	$B(S(X))$
X:PD	$CM(X) \geq 0.5$	$CM(X)$	R(CM(X))	X

Table 1: Conversion table

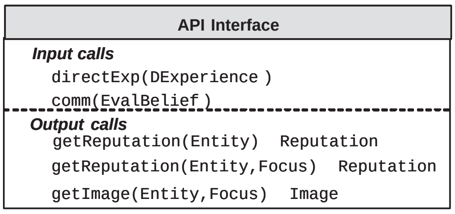

Figure 4: The first level of the API interface

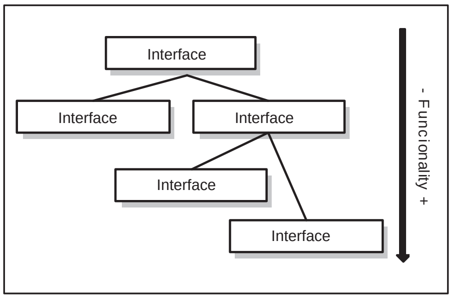

Figure 5: API interface hierarchy

5. THE API DESIGN

So far, we have defined an ontology that allows agents to communicate and reason on reputation concepts independently of their reputation model they are using. However, we need a module that translates the elements of the ontology to elements understandable by each particular reputation model, an ontology mapping. We defined then an API as an interface with a set of common operations whose inputs and outputs are elements of the ontology, and must be implemented for each particular model. For simple models, it will be enough to implement the operations for the first level of the API interface (see Figure 4). For more complex models, new operations may be needed to use all the expressiveness, getting a hierarchy of interfaces (see Figure 5).

- **Input calls**

  In the first level of the hierarchy (see Figure 4) we defined two functions needed to feed most of the trust and reputation systems that currently exist. A direct experience is the main source to build up an image. Then, an agent that interacts with another agent signing a contract would generate an outcome that would be the direct experience of the agent on this particular transaction. This fact will be notified to the reputation model used by the agent, if it is a subjective model, through the API function *directExp*. The other main way to feed reputation models is through communications from other entities. Then, when a communications is received the model will be fed using the API function *comm*. Notice that in principle, whatever evaluative belief can be communicated. The implementation of each API interface will be in charge of translating these entrances from elements of the ontology to elements understandable by the reputation model, and therefore, decide what means in each case the fact of communicating, for instance, an image, reputation or whatever evaluative belief.

- **Output calls**

  In this first level we decided to specify only three functions that give support to the most common queries that any agent may require from a reputation model (see Figure 4): knowing the reputation and the image a of certain entity. The reason we describe two versions of *getReputation* is due to the existence of some classical reputation models, like eBay [10], that use implicit context information, and in terms of the ontology means that there is no *Focus*. Again, here the API will query the reputation system in its own understandable language, that will respond with certain value that the same API will translate to the respective object of the ontology. Notice that the main point is that the decision making module of the agent always reason over the concepts described by the ontology. That would allow communication between agents that use different reputation models (if they have an implementation of the respective API).

6. IMPLEMENTATION AND EXAMPLES

We decided to implement several well known reputation and trust models and its respective API interfaces to prove the effectiveness of our approach. In this section we briefly describe these models and the first level of the API interface.

6.1 eBay model

eBay site [10] is one of the most concurred (if not the most) online marketplace in the world with more that 50 million registered users. As we stated above, eBay reputation model considers reputation as a public and centralized value that is not dependent on the context [7]. In this case, users can rate the seller they have bought items from after each transaction, with values of +1, 0 , -1. The reputation value of the sellers is calculated with the sum of all the ratings over the last six month.

- **API input calls**

  Taking into account that this is a centralized model, it is fed only with external ratings that in the ontology are communication of direct experiences. From our approach, the fact of rating with +1, 0, -1 a given transaction is a subjective evaluation of this transaction that coincide with the class *DirectExperience*. Then, the API needs to implement the function *comm(DExperience)*. The representation value we have chosen in this case is the discrete set, with the following equivalence: $\{-1 \leftrightarrow VB,\ 0 \leftrightarrow N,\ +1 \leftrightarrow VG\}$

- **API output calls**

  The model provides to the users the plain sum of all the ratings for each seller with a colored stars system. According to the definition in the theory of reference used in this work (see Section 3), this value can be seen as sellers' reputation, and therefore, the API function to be implemented should be *getReputation(Entity)* $\rightarrow$ *Reputation* The best way to represent the value of the Reputation object clearly is the bounded real. Since eBay punctuation goes from 0 to 100000, a simple normalized transformation to the interval [0,1] seems to be enough. However, notice that the colored stars representation does not follow a linear curve. From a semantic point of view and in our value representation, 0 means very bad reputation, 0.5 neutral reputation, and 1 very good reputation, with a totally linear function. In eBay, having more that 10 point is already considered a good reputation. The next step in the scale is more than 100 points (with a different colored star), and the next is more than 500. In conclusion there is no lineal relation between the punctuation and the semantic representation of the stars. Then, it is necessary a transformation from the ontology representation value to the eBay scale. A possible transformation function is described in the following equation:

  $$H : [0, 100000] \rightarrow [0, 1] \tag{19}$$

  $$H(X) = \begin{cases} 0 & \text{if } X < 10; \\ 1 & \text{if } X > 100000; \\ \frac{log(X)-0.5}{8} + 0.5 & otherwise. \end{cases} \tag{20}$$

  In this case, no strength value is taken into account, considering every reputation value with the maximum possible strength.

6.2 Sporas model

This model [27] considers reputation, like eBay model, as a public and centralized value. In this case though, only the most recent ratings between two users are considered. Furthermore, users with very high reputation values experience much smaller changes than with low reputation after each update due to the aggregation function. In fact, the model is parameterizable with different parameters such like the range of the reputation values or the number of ratings to consider for the calculus. For the integration of the model with the API, we need to know how reputation values are presented and its semantics, as well as the rating values. Like in eBay, the system is fed by communication of direct experiences of the users rating sellers. Then the API function *comm(DExperience)* needs to be implemented. Because users query the system asking for reputation values the function *getReputation(Entity)* $\rightarrow$ *Reputation* needs to be implemented as well. In both cases, the representation value of the evaluation clearly is the bounded real, since the reputation value as well as the rating measures are numbers belonging to the interval defined by the range, following a totally linear curve: that is, the minimum value of the range is the worst reputation value (and rating value) and the maximum value of the range is the best one, with a linear gradient from the minimum to the maximum. Then a simple normalized transformation can be done.

6.3 Abdul-Rahman and Hailes Model

This model [1] uses the term *trust*, understanding it as a distributed and subjective value. It means that every agent has its own reputation system as a submodule of the architecture of the agent. In this case, social evaluations take into account the context, the focus element in terms of the ontology. The model is fed by two sources: direct experiences and third party communications of direct experiences. The representation of the evaluations is done in terms of the discrete set {*vt (very trustworthiness), t (trustworthiness), u (untrustworthiness), vu (very untrustworthiness)*}. Then, for each agent and context the system keeps a tuple with the number of past own experiences or communicated experiences in each category. For instance, agent A may have a tuple of agent B as a seller like $(0, 0, 2, 3)$, meaning that agent A has received or experienced 2 results as untrustworthiness and 3 as very untrustworthiness. Finally the *trust* value is computed taking the maximum of the tuple values. In our example for agent A, agent B as a seller would be very untrustworthy. In case of tie between vt and t and between u and vu the system gives the values U^+ (mostly trustworthy) and U^- (mostly untrustworthy) respectively. In any other tie case the system returns U^0 (neutral).

- **API input calls**

  As we described, this model is fed by communications of direct experiences from other agents, and direct experiences of the same agent. In this case then we need to implement the functions *comm(DExperience)* and *directExp(DExperience)*. Since in the model direct experiences are evaluated with one of the four categories, {vt,t,u,vu} seems logical to represent it in the ontology with the discrete set, using the following equivalences: vt $\leftrightarrow$ VG, t $\leftrightarrow$ G, u $\leftrightarrow$ B, vu $\leftrightarrow$ VB.

- **API output calls**

  The *trust* measure that the model provides, in terms of the ontology, is close to the concept of *image*, because agents accept as true the measure. Then, we need the function *getImage(Entity,Focus)* $\rightarrow$ *Image*. Here, we decided to use as a representation value of the image

[7] In fact, the context is determined by the environment where this model is used: an e-Auction market

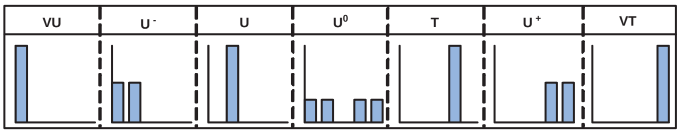

Figure 6: Transformation of Abdul-Rahman and Hailes model trust values to probabilistic distribution(PD)

evaluation a probabilistic distribution type, since the semantics of U^+, U^+ and U^0 involve more than one *trust* measure[8]. In this model, when the answer is *vu* means that for sure the agent is very untrustworthy, meaning that with a probability of 1 the agent is very bad (in terms of the ontology). In the same way when the model achieves the value U^- means that with the same probability, the agent is untrustworthy or very untrustworthy, that can be translated into probabilistic distribution as having a probability of 0.5 of being VB and 0.5 of being B. The total transformation table is showed in Figure 6.

6.4 Repage

Repage [19] is a distributed computational system based in the cognitive theory of reputation described in [9]. Consequently the main elements of the presented ontology appears and its semantics is very similar. In this case, the model keeps as well a difference between Image and Reputation. The memory consists of a set of connected predicates (beliefs) conceptually organized in layers, where in the bottom we find non evaluated predicates, like communications from other partners and contracts (before a direct interaction) and fulfillments (the result of the direct interaction). From communicated reputations of the same target and context, a shared voice predicate is generated that when it is strong enough, ends up as a reputation predicate. Similarly, communicated images and third party images[9] of the same target and context generate what is called a shared evaluation predicate, that with the outcome predicate obtained from direct interactions (through the subjective evaluation of fulfillments) may generate an image predicate over the target in the context. The representation value of social evaluations uses the probabilistic distribution we have described in this paper together with a strength value, a real bounded number. We refer to [19] for a more detailed description of all these concepts and the aggregation functions used to add up different probabilistic distribution values.

- **API input calls**

  This model is fed by communications of images, reputations and third party images, and by direct experience using the mechanism of signing a contract, generating a fulfillment of the transaction and evaluating it generating an outcome. Communication issues require the implementation of the functions *comm(Image)* and *comm(Reputation)*. Notice that third party images are images where the source and the gossiper are not the same. However, the API function *comm* always carries the origin of the communication, the gossiper. If it is not the same that the source of the evaluation of the image, it is a third party image.

  Direct interaction requires a small change. Notice that for each interaction, there is a contract predicate, a fulfillment predicate and an outcome predicate, being basically the goodness or badness of the fulfillment respect the original contract. This calculus is done inside the model. However, in order to use this ontology, the concept of *outcome* has to be seen as a direct experience object, and therefore, needs to be calculated outside the model, in the agent part. The situation is not critical though. Already in Repage, outcome predicates are totally context dependent, and for each situation have to be implemented. Then the API function *directExp(DExperience)* will represent the insertion of an outcome in the Repage system. Because this model uses already a probabilistic distribution representation of the evaluations, there is no reason to not use the same type in the direct experience object.

- **API output calls**

  The system basically provides images and reputations of a given agent in certain context. Then, the main API operations to implement are *getImage(Entity,Focus)* $\rightarrow$ *Image* and *getReputation(Entity,Focus)* $\rightarrow$ *Reputation*. The mapping from the model to the elements of the ontology is direct, besides minor structural points. Furthermore, there is no need to readjust evaluations values representation, since we can keep using the same probabilistic distribution.

6.5 Conclusions on the API implementation part

What we have showed in this section is a small set of representative reputation models and its mapping to the ontology through an implementation of a set of operations from an API interface. In this work we only implemented the first level of the API hierarchy of each model, but more levels could be implemented in function of the necessities and possibilities of each model. For instance, Repage model [19] incorporates the concept of *Shared Evaluation* (see subsection 6.4) that can be mapped to the class *SharedImage*. In this second level of the Repage API hierarchy we could incorporate functions to access to these elements.

7. CONCLUSIONS AND FUTURE WORK

In this work we have presented an implementable solution for the communication of social evaluations between agents using different reputation models that is required in multiagent paradigms. This solution offers an ontology of reputation that agents can share as a common *language* to exchange reputation concepts. The communication through a common ontology requires an ontology mapping linking the own ontology used by each reputation model with this common one. We implemented this mapping using a hierarchy

[8]For instance, U^- is done when there is a tie between *vu* and *u*, meaning that both options have a 50% of probability

[9]A third party image communication is a communicated image where the source of the image is not the source of the communication. For instance, agent A may communicate to somebody that for agent C, agent D is BAD

of API interfaces that offer set of primitive function to be implemented for every specific model.

This approach allows to have a library of APIs that users could use when designing decision making modules without having to think in the peculiarities of each reputation model, and having total freedom for using different reputation models in different agents that participate in the same society. As a future work, we plan to study also the use of other entropy measures like differential entropy or relative entropy and mutual information for the calculation of the CU.

Another line of future research is the use of what is called dynamic ontology alignment. Notice that the presented ontology is static. It means that for every new reputation model we would need a new API implementation and, even when currently there is no model that could use all the expressiveness of this ontology (except may be Repage [19]), it is not a guarantee in the future. The main idea behind the dynamic ontology alignment is that by using a very basic but commond set of elements that two agents share, they can build at runtime a common ontology specific for that interaction. In our domain, it would mean they were able to understand, for instance, the concepts of goodness or badness of the other agents. We expect to design mechanisms based on ontology alignment theory to achieve this objective.

8. ACKNOWLEDGMENTS

This work was supported by the European Community under the FP6 programme (eRep project CIT5-028575, OpenKnowledge project FP6-027253 and SocialRep project MERG-CT-2005/029161). Jordi Sabater-Mir enjoys a Ramon y Cajal contract from the Spanish Government.

9. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the Hawaii's International Conference on Systems Sciences, Maui, Hawaii*, 2000.

[2] Amazon. *Amazon Auctions*. http://auctions.amazon.com, 2002.

[3] D. B. Bromley. *Reputation, Image and Impression Management*. John Wiley & Sons, 1993.

[4] V. Buskens. The social structure of trust. *Social Networks*, (20):265—298, 1998.

[5] J. Carbo, J. Molina, and J. Davila. Trust management through fuzzy reputation. *Int. Journal in Cooperative Information Systems*, pages in–press, 2002.

[6] J. Carter, E. Bitting, and A. Ghorbani. Reputation formalization for an information-sharing multi-agent sytem. *Computational Intelligence*, 18(2):515—534, 2002.

[7] S. J. Casare and J. S. Sichman. Towards a functional ontology of reputation. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems, Utrecht, The Netherlands*, pages 505–511. ACM Press, 2005.

[8] M. Celentani, D. Fudenberg, D. K. Levine, and W. Psendorfer. Maintaining a reputation against a long-lived opponent. *Econometrica*, 64(3):691—704, 1966.

[9] R. Conte and M. Paolucci. *Reputation in artificial societies: Social beliefs for social order*. Kluwer Academic Publishers, 2002.

[10] eBay. *eBay*. http://www.eBay.com, 2002.

[11] eRep. *eRep Wiki*. http://megatron.iiia.csic.es/mediawiki/index.php, 2006.

[12] eRep. *eRep:Social Knowledge for e-Governance*. http://megatron.iiia.csic.es/eRep, 2006.

[13] B. Esfandiari and S. Chandrasekharan. On how agents make friends: Mechanisms for trust acquisition. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 27—34, 2001.

[14] D. Hume. *A Treatise of Human Nature (1737)*. Oxford: Clarendon Press, 1975.

[15] M. Karlins and H. I. Abelson. *Persuasion, how opinion and attitudes are changed*. Crosby Lockwood & Son, 1970.

[16] R. Marimon, J. P. Nicolini, and P. Teles. Competition and reputation. In *Proceedings of the World Conference Econometric Society, Seattle*, 2000.

[17] OnSale. *OnSale*. http://www.onsale.com, 2002.

[18] Plato. *The Republic (370BC)*. Viking Press, 1955.

[19] J. Sabater, M. Paolucci, and R. Conte. Repage: Reputation and image among limited autonomous partners. *Journal of Artificial Societies and Social Simulation*, 9(2), 2006.

[20] J. Sabater and C. Sierra. Regret: A reputation model for gregarious societies. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 61—69, 2001.

[21] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societites. *Applied Artificial Intelligence*, (Special Issue on Trust, Deception and Fraud in Agent Societies), 2000.

[22] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-02), Bologna, Italy*, pages 288—293, 2002.

[23] C. Shannon. A mathematical theory of communication. *Bell System Thecnical Journal*, 27:379—423, 1948.

[24] A. Valente. *Legal Knowledge Engineering - A modelling approach*. IOS Press, 1995.

[25] L. Vercouter, S. Casare, J. Sichman, and A. Brando. An experience on reputation models interoperability based on a functional ontology. In *IJCAI '07: Proceedings of the Twentieth International Join COnference on Artificial Inteligence, Hyderabad, India*, pages 617—622, 2007.

[26] B. Yu and M. P. Singh. Towards a probabilistic model of distributed reputation management. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 125—137, 2001.

[27] G. Zacharia. Collaborative reputation mechanisms for online communities. Master's thesis, Massachusetts Institute of Technology, September 1999.

APPENDIX

A. CALCULUS OF CU

A.1 $CU(BO, DS) = 1.29$

Considering True as t and False as f:

$$CU(BO, DS) = H(DS|BO) \tag{21}$$

Knowing that $p(BO = t) = p(BO = f) = \frac{1}{2}$ we can write that

$$CU(BO, DS) = \frac{1}{2}H(DS|BO = t) + \frac{1}{2}H(DS|BO = f) \tag{22}$$

At this point, when $BO = t$ and following our semantic interpretation we know that it may refer to one value of the set $\{N, G, VG\}$, and if $BO = f$ of the set {VB,B}. Then $P(DS = \{VB\}|BO = f) = P(DS = \{B\}|BO = f) = 1/2$ (zero in other values of DS) and $P(DS = \{N\}|BO = t) = P(DS = \{G\}|BO = t) = P(DS = \{VG\}|BO = t) = 1/3$ (zero in other values of LL). Then, following the previous equations and developing the entropy formula we have that

$$H(DS|BO = t) = -3\frac{1}{3}\log(\frac{1}{3}) \approx 1.58 \tag{23}$$

$$H(DS|BO = f) = -2\frac{1}{2}\log(\frac{1}{2}) = 1 \tag{24}$$

finally, computing the equation 22 we have

$$CU(BO, DS) = 0.79 + 0.5 = 1.29 \tag{25}$$

A.2 $CU(BO, RE) = 5.64$

Here, knowing that $BO = t$ our semantic indicates that as a real, it could be a value from 0.50 and 1, then $\forall_{i\in[0,1]}p(RE = i|BO = t) = p(RE = i|BO = f) = 1/50$ and therefore,

$$H(RE|BO = t) = H(RE|BO = f) = -50\frac{1}{50}\log(\frac{1}{50}) \approx 5.64 \tag{26}$$

$$CU(BO, RE) = 5.64 \tag{27}$$

A.3 $CU(BO, PD) = 21.19$

Having in mind the total number possible elements in PD (see equation 17), we know that $BO = t$ implies that whatever representation of PD will tend towards a good evaluation, it means that the probability of being good is higher that the opposite. That eliminates exactly 50% of all the representations, and therefore

$$\forall_{i\in PD}p(PD = i|BO = t) = p(PD = i|BO = f) = \frac{2}{A} \tag{28}$$

$$H(PD|BO = t) = H(PD|BO = f) = -\frac{A}{2}\frac{2}{A}\log(\frac{2}{A}) \approx 21.19 \tag{29}$$

$$CU(BO, PD) = 21.19 \tag{30}$$

A.4 $CU(DS, RE) = 4.32$

Following the same reasoning:

$$CU(DS, RE) = H(RE|DS) \tag{31}$$

$$CU(DS, RE) = \sum_{i\in\{vb,b,n,g,vg\}} \frac{1}{5}H(RE|DS = i) \tag{32}$$

Notice that in this case, the difference between a Real and DS is that the first is continuous and the second discrete. Then, dividing the $[0, 1]$ interval into five identical parts, and assigning each of them into a value of DS we have the problem almost done. In this situation, each value of DS correspond to a 20 values of Real, and therefore,

$$\forall_{i\in\{vb,b,n,g,vb\}}\forall_{j\in[0,1]}p(RE = j|DS = i) = \frac{1}{20} \tag{33}$$

Then,

$$\forall_{i\in\{vb,b,n,g,vb\}}H(RE|DS = i) = -20\frac{1}{20}\log(\frac{1}{20}) \approx 4.32 \tag{34}$$

and then,

$$CU(DS, RE) = 4.32 \tag{35}$$

A.5 $CU(DS, PD) = 19.89$

The key in all the calculus is in the fact that each element of DS may correspond to a set of elements of PD whose center of mass is included in the interval corresponding to the function defined in R'. In the same way we have discretized the interval $[0, 1]$ in five parts, for each of these intervals we have a total of $\frac{A}{5}$ elements of PD with a center of mass that points inside the interval. Therefore, we can establish the following statement:

$$\forall_{i\in\{vb,b,n,g,vb\}}\forall_{j\in PD}p(PD = j|DS = i) = \frac{5}{A} \tag{36}$$

and,

$$\forall_{i\in\{vb,b,n,g,vb\}}H(PD|DS = i) = -\frac{A}{5}\frac{5}{A}\log(\frac{5}{A}) \approx 19.89 \tag{37}$$

then,

$$CU(DS, PD) \approx 19.89 \tag{38}$$

A.6 $CU(RE, PD) = 15.55$

Following the same reasoning than in the previous point, the number of elements of PD whose center of mass is the one being converted is approximately $\frac{A}{100}$, and therefore,

$$\forall_{i\in[0,1]}\forall_{j\in PD}p(PD = j|RE = i) = \frac{100}{A} \tag{39}$$

and,

$$\forall_{i\in\{vb,b,n,g,vb\}}H(PD|RE = i) = -\frac{A}{100}\frac{100}{A}\log(\frac{100}{A}) \approx 15.55 \tag{40}$$

then,

$$CU(RE, PD) = 15.55 \tag{41}$$