Andreas Schlosser, Marco Voss and Lars Brückner (2006)

On the Simulation of Global Reputation Systems

Journal of Artificial Societies and Social Simulation vol. 9, no. 1
<http://jasss.soc.surrey.ac.uk/9/1/4.html>

For information about citing this article, click here

Received: 13-Mar-2005 Accepted: 22-Jul-2005 Published: 31-Jan-2006

Abstract

Reputation systems evolve as a mechanism to build trust in virtual communities. In this paper we evaluate different metrics for computing reputation in multi-agent systems. We present a formal model for describing metrics in reputation systems and show how different well-known global reputation metrics are expressed by it. Based on the model a generic simulation framework for reputation metrics was implemented. We used our simulation framework to compare different global reputation systems to find their strengths and weaknesses. The strength of a metric is measured by its resistance against different threat-models, i.e. different types of hostile agents. Based on our results we propose a new metric for reputation systems.

Keywords:
Reputation System, Trust, Formalization, Simulation

Introduction

1.1

Reputation systems are an important building block for achieving trust within large distributed communities, especially when mutually unknown agents engage in ad-hoc transactions. Based on the knowledge of past behavior it is up to each agent to form his opinion on his potential transaction parter.

1.2

This paper is focused on comparing the effectiveness of different methods to compute reputation based on past behavior. A system is effective if it perceives and penalizes agents who try to undermine the community by cheating on their transaction partners. The effectiveness of the reputation systems is evaluated by simulation. Different threat types of misbehaving agents are modeled and for each system it is checked to which degree the system can stand the attack. Other important implementational aspects of reputation systems, including performance, quality of service, protection against hacker attacks, and privacy of transaction data are out of the scope of our analysis and are not discussed here.

1.3

The contributions of this paper are to present a formal model of reputation and an overview of metrics used in different global reputation systems. Based on the formal model a simulation framework was developed to test and compare different reputation metrics. We summarize our simulation results and present a combined metric, called *BlurredSquared*, as an optimization.

1.4
The remainder of the paper is organized as follows. In section 2 we give an introduction into the field of reputation systems and present a formal model for metrics in reputation systems. Section 3 describes how our model is applied to different types of metrics. Section 4 deals with the simulation. There we present our simulation framework, introduce the agent models which are used for simulation, and evaluate the simulation results. Related work is discussed in section 5. We conclude the paper and give an outlook on future work in section 6.

Reputation Systems

2.1
Reputation is a subject of research in a lot of disciplines. There are many different definitions of the terms *reputation* and *trust* and no accepted common model of reputation exists. Mui et al. (2002) have provided an overview of the different notions these terms have in various disciplines and have developed a typology of kinds of reputation. We do not repeat this discussion here but use an intuitive definition of reputation and trust:

> Reputation is the collected and processed information about one entity's former behavior as experienced by others. "Trust is a measure of willingness to proceed with an action (decision) which places parties (entities) at risk of harm and is based on an assessment of the risks, rewards and reputations associated with all the parties involved in a given situation." (Mahoney 2002)

2.2
A *rating* is a single opinion about the outcome of a transaction. Reputation systems (Resnick and Zeckhauser 2000) monitor an agent's behavior by collecting, aggregating and distributing such feedback. Conceptually, a reputation system consists of the following components and actors (see figure 1): The target of a rating is called *ratee*. The *collector* gathers ratings from agents called *raters*. This information is processed and aggregated by the *processor*. The algorithm used by the processor to calculate an aggregated representation of an agent's reputation is the *metric* of the reputation system. The *emitter* makes the results available to other requesting agents.

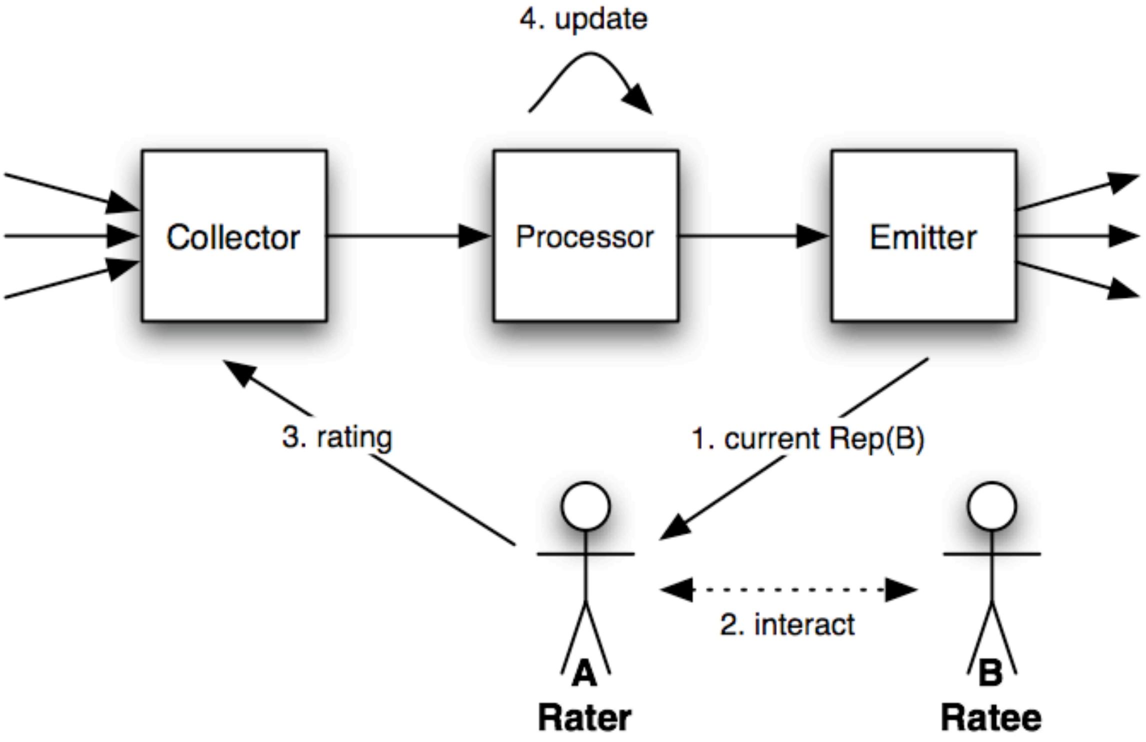


Figure 1: Architecture of a reputation system

2.3
A reputation system is *centralized* if only a single or a small number of processing entities exist.

It is *distributed* when every agent calculates reputation values about his partners or neighbors by aggregating all available information. Within a *global* reputation system, there is only a single reputation value per agent. *Local* reputation systems provide different reputation values between different sets of agents. An agent can have different reputation values depending on the requesting agent that issues the query. Mui et al. (2002) call this *personalized* reputation. Local can also mean that not all rating information is accessible everywhere. For instance in a distributed system with propagation there is some delay before a submitted rating is available at all nodes. The range of propagation may be restricted to a certain number of hops like in Damiani, di Vimercati, Paraboschi, Samarati, and Violante (2002). Another example of local reputation is the approach to take the underlying social network into account (Huynh, Jennings, and Shadbolt 2004; Sabater and Sierra 2001; 2002; Zacharia and Maes 2000). Reputation systems can also be distinguished by their representation of some reputation value. Most reputation systems use *definite* values, but there are some systems using so called *fuzzy* values as a representation for reputation (Carbo, Molina, and Davila 2003). However, in this paper we focus on global reputation systems with definite reputation values.

2.4

Reputation systems have to deal with a number of threats. Dellarocas (2000) describes the problems of unfair ratings and discrimination. Important is the 'ballot stuffing' attack where a group of agents collude to give one member high ratings. In some systems there is an incentive for a bad behaving agent to switch its identity after some time and to start over again (Friedman and Resnick 2001). This happens if the reputation of new members is better than the minimum possible reputation in the system. On the other hand it must not be too difficult for new members to enter the system. Consequently, the initial reputation of an agent has to be chosen carefully.

Context

2.5

Reputation is context dependent. In general reputation earned in one context cannot be applied in another context. For instance, a reputation for being an expert in computer science says nothing about being a trustworthy customer. As a consequence reputation cannot be easily aggregated into a single number. Additionally, a rating may contain different aspects. For customers of an online shop not only the quality and price of a product are important but also the delivery time and after sales services. Reputation is more a vector than a scalar value where each entry represents the reputation with respect to a well specified context.

2.6

Most existing reputation systems ignore this fact or are restricted to a single and simple context. eBay's feedback system (eBay Homepage 2004; Resnick and Zeckhauser 2002) – as a well studied example – does not distinguish between ratings for buyers or sellers, although these roles are very asymmetric. The seller of a product has much more information about its quality than the buyer and will receive payment before shipment in most cases. Consequently, the risk of a seller is limited. Additionally, the value of the traded goods is ignored by eBay's feedback system. This allows to build up a high reputation by selling goods with low value.

2.7

The context of a transaction between two agents contains information about its circumstances and the environment. Examples are topic, time, quality, value or role of the participants. Mui et al. (2001) define the context based on a number of attributes which are present or not.

2.8

If there are two contexts which are compatible to some degree, a mapping between the reputation information should be possible. However, it is not clear when and how this transfer is possible. Approaches that make use of ontologies (Maximilien and Singh 2003) may provide a solution to this problem.

2.9

In this paper we use an abstract scenario of a space where agents provide homogeneous symmetric services to each other. Thus we do not try to present a detailed model of context here. However, context and mapping between different contexts should be a topic of future research.

Formal Model

2.10

We now present a formal model for reputation that is used throughout this paper. It is not the aim of this model to include every aspect of a reputation system. Especially, the flow of information, the location of processing and storage, the query mechanism and incentives to give feedback are not part of the model. The model provides an abstract view of a reputation system that allows the comparison of the core metrics of different reputation systems.

2.11

According to our definition of reputation a transaction between two peers is the basis of a rating. An agent cannot rate another one without having had a transaction with him.

2.12

A is the set of agents. C is the context of a transaction. In the following we assume a simple uniform context and set $C = T \times V$ where $T = \{0, 1,..., t_{now}\}$ is the set of times and V is the set of transaction values. We define E as the set of all encounters between different agents that have happened until now. An encounter contains information about the participating peers and the context:

$$E = \{(a, b, c) \in A \times A \times C \mid a \neq b\}$$

2.13

A rating is a mapping between a target agent $a \in A$ and an encounter $e \in E$ to the set of all possible ratings Q:

$$\rho : A \times E \rightarrow Q \cup \{ \propto \}$$

where $\propto$ means undefined. Depending on the system Q can have different shapes. In the simple case Q is a small set of possible values: $Q_{ebay} = \{ -1, 0, 1\}$ or an interval $Q_i = [0, 1]$. Complex schemes like in Whitby, Jøsang, and Indulska (2004) are possible, too: $Q_r = \mathbb{R}^{+}{}_{0} \times \mathbb{R}^{+}{}_{0}$.

2.14

E_a represents the subset of all encounters in which a has participated and received a rating:

$$E_a := \{e \in E \mid (e = (a, \cdot, \cdot) \vee e = (\cdot, a, \cdot)) \wedge \rho(a, e) \neq \propto \}$$

All encounters between a and b with a valid rating for a are:

$$E_{a,\,b} := \{e \in E_a \mid e = (a, b, \cdot) \vee e = (b, a, \cdot)\}$$

Furthermore we define

$$E_a{}^{*} := \bigcup_{b \in A} \{e \in E_{a,\,b} \mid \rho(a, e) \neq \propto \wedge \tau(e) = max\}$$

as the subset of all most recent encounters between *a* and other agents. $\tau(e)$ gives the time of encounter *e*. $\nu(e)$ gives the value of the transaction *e*.

2.15

We define $\overline{E_a}$ and $\overline{E_a^*}$ to be time-sorted lists of the sets E_a and E_a^*. The Operator # gives the size of a set or a list. We get the elements of a list *L* through *L*[*i*] where $i \in \{1,...,\#(L)\}$. An encounter between *a* and *b* at the specific time *t* is $e_{a,\ b}^{t} \in E_{a,\ b}$.

2.16

The reputation of an agent $a \in A$ is defined by the function $r : A \times T \rightarrow R$. The properties of *R* have already been discussed in the previous section. In most cases it is a subset of $\mathbb{R}$. We use $r(a) := r(a, t_{now})$ for short. r_0 describes the initial reputation $r(a, 0)$ of a new agent. A complete metric $\mathscr{M}$ is defined as $\mathscr{M} = (\rho, r, Q, R, r_0)$, or for short by the pair $\mathscr{M} = (\rho, r)$.

2.17

Please note, that the model developed in this section fits on reputation systems with distinct reputation values. It has to be adapted slightly to work with fuzzy systems like presented in Carbo, Molina, and Davila (2003)

Metrics in Reputation Systems

3.1

Within the model given above, a reputation system is described by its specific metric. This allows us to compare different systems by measuring the computed reputation values within certain communities. The next sections give an overview on the metrics that have been subject of our simulations. Most of the systems have been analyzed with and without taking the value of a transaction into account, and with or without multiple ratings per agent pair. The figures show a characteristic graph for each system. The horizontal axis is the time axis, and the vertical axis represents the reputation computed by the specific metric, based on the collected ratings so far. The depicted reputation is distinguished according to several simulated agent types, as described later in section 4. More details about how these figures are generated will be explained in this section, too.

Accumulative Systems

3.2

If a system accumulates all given ratings to get the overall reputation of an agent we call it an accumulative system. The well known feedback system of eBay (eBay Homepage 2004) is an example. We have implemented simulations of this kind of systems with and without considering the transaction values and multiple ratings from the same agent. The possible ratings are $\rho : A \times E \rightarrow \{-1, 0, 1\}$. The basic idea of these metrics is, that the more often an agent behaves in a good way the more sure can the others be, that this agent is an honest one. It is accepted, that an agent can iron out some bad ratings he received just by further good transactions.

3.3

However, these systems also allows an agent to behave bad in a certain fraction of transactions and still to improve its overall reputation if this fraction is small enough. Recently, eBay has updated its feedback system to also include the percentage of positive transactions in the detailed member profile.

3.4

In the *eBay*-system itself, no transaction values and multiple ratings are considered. The reputation of an agent $a \in A$ computes with:

$$r(a) = \sum_{e \in E_a^*} \rho(a, e) \tag{\textit{eBay}}$$

With consideration of transaction values, the reputation in the *Value*-system computes with:

$$r(a) = \sum_{e \in E_a^*} \rho(a, e) \cdot \nu(e) \tag{\textit{Value}}$$

Adding multiple ratings, we get the *SimpleValue*-system with the following reputation computation:

$$r(a) = \sum_{e \in E_a} \rho(a, e) \cdot \nu(e) \tag{\textit{SimpleValue}}$$

The *Simple*-system considers multiple ratings, but no transaction values. Thus the reputation for an agent $a \in A$ computes with:

$$r(a) = \sum_{e \in E_a} \rho(a, e) \tag{\textit{Simple}}$$

Figure 2 shows the development of the reputation in the *Value*-system.

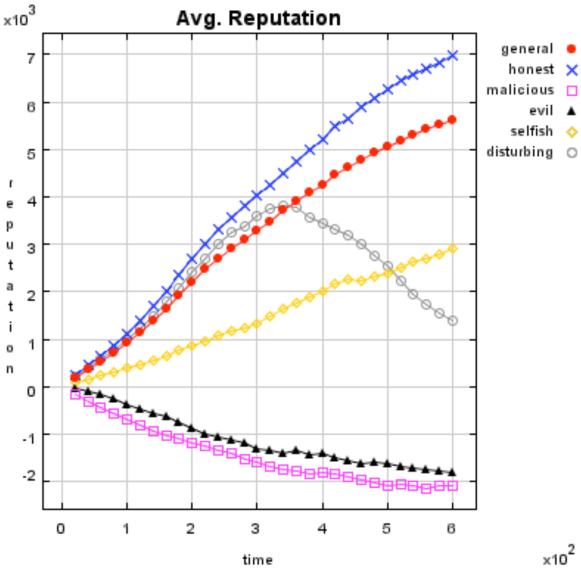


Figure 2: The *Value*-system

3.5

Dellarocas has done a theoretical analysis of accumulative systems in Dellarocas (2003). His setup is $Q = \{-1, 0\}$ and

$$r(a, t) = \sum_{e \in D_t} \rho(a, e)$$

where $D_t \subset E_a$ has n elements. This means that only negative ratings are recognized. In every round a random element $e_r \in D_t$ is replaced by the newest encounter: $D_{t+1} := D_t \setminus \{e_r\} \cup \{e_{t+1}\}$. If $n = 1$ we have a system as described in 3.10. Some of Dellarocas' results are reproduced by our simulations as we will describe in the evaluation.

Average Systems

3.6

This kind of reputation system computes the reputation for an agent as the average of all ratings the agent has received. This average value is the global reputation of this agent. An example can be found in Jurca and Faltings (2003). The idea of this metric is, that agents behave the same way most of their lifetime. Unusual ratings have only little weight in the computation of the final reputation. This could be used to place some bad transactions intentionally by bad agents. The simulated systems use $\rho : A \times E \rightarrow \{-1, 0, 1\}$. The computation is done is several ways, with or without considering the transaction value and multiple ratings.

3.7

The reputation of an agent $a \in A$ in the *Average*-system without considering multiple ratings and transaction values is:

$$r(a) = \frac{\sum_{e \in E_a^*} \rho(a, e)}{\#(E_a^*)} \qquad (Average)$$

The reputation of an agent $a \in A$ in the *AverageSimple*-system without considering the transaction value but including multiple ratings is:

$$r(a) = \frac{\sum_{e \in E_a} \rho(a, e)}{\#(E_a)} \qquad (AverageSimple)$$

The reputation of an agent $a \in A$ in the *AverageSimpleValue*-system including multiple ratings and the transaction value is:

$$r(a) = \frac{\sum_{e \in E_a} \rho(a, e) \cdot v(e)}{\#(E_a)} \qquad (AverageSimpleValue)$$

The reputation of an agent $a \in A$ in the *AverageValue*-system including the transaction value but not considering multiple ratings is:

$$r(a) = \frac{\sum_{e \in E_a^*} \rho(a,e) \cdot v(e)}{\#(E_a^*)} \qquad (AverageValue)$$

Figure 3 shows the development of the reputation in the *AverageValue*-system.

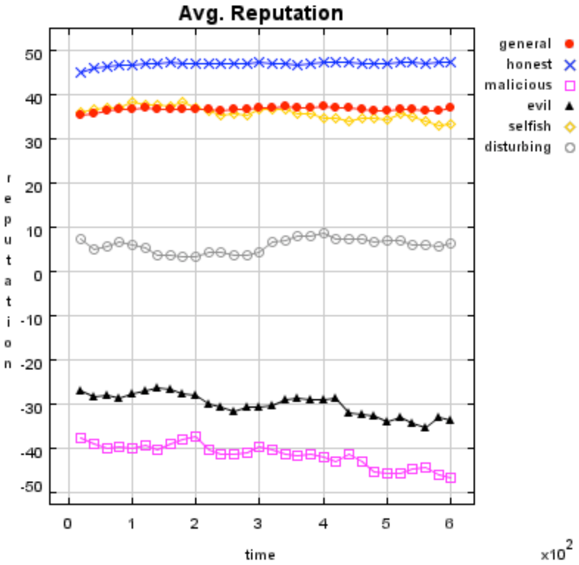


Figure 3: The *AverageValue*-system

Blurred Systems

3.8 These reputation systems compute a weighted sum of all ratings. The older a rating is, the less it influences the current reputation. An approach in a peer-2-peer environment is described in Selçuk, Uzun, and Pariente (2004), a metric with an unspecified time-dependent weight-function is used in Huynh, Jennings, and Shadbolt (2004). A blurred system is implemented with and without considering transaction values for the simulation. Multiple ratings are always considered in this system. This is no problem, because older ratings have a lower weight. Groups cannot manipulate their reputation by giving each other high ratings. Possible ratings are $\rho : A \times E \rightarrow \{-1, 0, 1\}$. This metric is based on the observation that agents do change their behavior during their lifetime. The assumption is, that they will behave more probably like they did in their most recent transactions than they did in transactions long ago in the past.

3.9 The reputation of an agent $a \in A$ without considering transaction values is:

$$r(a) = \sum_{i=1}^{\#(\overline{E_a})} \frac{\rho(a, \overline{E_a}[i])}{\#(\overline{E_a}) - i + 1} \qquad (Blurred)$$

and with consideration of transaction values:

$$r(a) = \sum_{i=1}^{\#(\overline{E_a})} \frac{\rho(a, \overline{E_a}[i]) \cdot v(\overline{E_a}[i])}{\#(\overline{E_a}) - i + 1} \qquad (BlurredValue)$$

Figure 4 shows the development of the reputation in the *Blurred*-system.

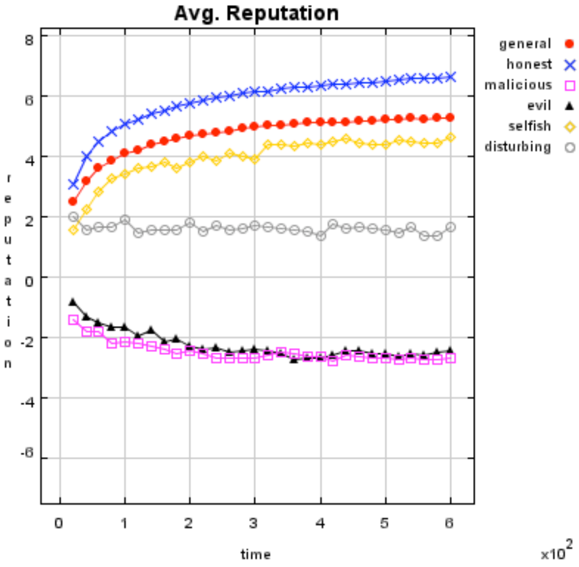


Figure 4: The *Blurred*-system

OnlyLast Systems

3.10

Even if only the most recent rating posted for an agent is regarded, the resulting reputation system is working. Dellarocas has formulated the same result in his theoretical work (Dellarocas 2003). This system is implemented in the simulation with and without consideration of transaction values, ratings are $\rho : A \times E \rightarrow \{-1, 0, 1\}$. This is an extreme variation of the *Blurred* system. Here we expect an agent to behave like he did last time, no matter what he did before.

3.11

Without considering transaction values in the *OnlyLast*-system the reputation of an agent $a \in A$ is:

$$r(a) = \rho(a, \overline{E_a}[\#(\overline{E_a})]) \qquad (OnlyLast)$$

With consideration of the transaction value in the *OnlyLastValue*-system the reputation of an agent $a \in A$ is:

$$r(a) = \rho(a, \overline{E_a}[\#(\overline{E_a})]) \cdot \nu(\overline{E_a}[\#(\overline{E_a})]) \qquad (OnlyLastValue)$$

Figure 5 shows the development of the reputation in the *OnlyLast*-system.

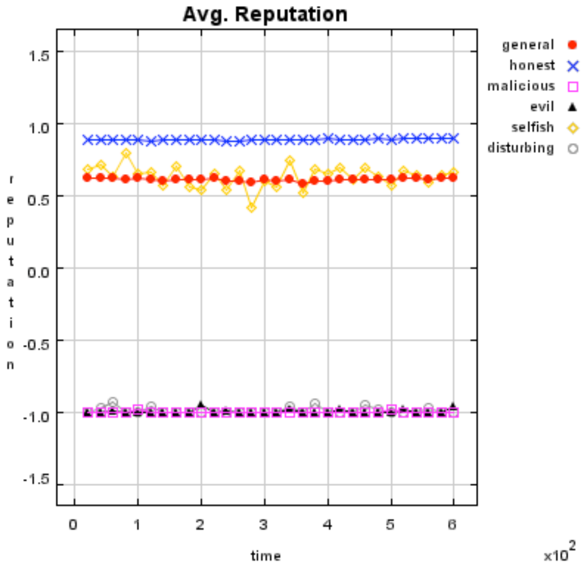


Figure 5: The *OnlyLast*-system

EigenTrust System

3.12

This system combines the local reputation values of each agent iteratively to a global reputation (Kamvar, Schlosser, and Garcia-Molina 2003). This is done by modifying a target agent's reputation values stored locally at one agent *a* by the opinions of surrounding agents. These opinions are weighed according to the local reputation values *a* has about its neighbors. During this process the individual reputations are iteratively accumulated to one single global reputation for each agent. This system is a special instance of the metric described in Xiong and Liu (2004). In this metric the computed reputation depends on the ratings, the reputation of the raters, the transaction context (e.g. transaction value), and some community properties (e.g. the total amount of given ratings). Another example is the Sporas-system in the next section 3.14, where each rating is weighted by the reputation of the rater.

3.13

The algorithm for the *EigenTrust*-system is described below. Legal ratings are $\rho : A \times E \rightarrow \{-1, 1\}$. First we have to build a reputation matrix *M*, where (m_{ij}) contains the standardized sum of ratings from Agent *i* for Agent *j*:

$$(m_{ij}) = \frac{\max\left(\sum_{e \in E_{i,j}} \rho(j,e), 0\right)}{\sum_j \max\left(\sum_{e \in E_{i,j}} \rho(j,e), 0\right)}$$

$$\vec{t}^{(0)} = \begin{pmatrix} 1/\#(A) \\ \vdots \\ 1/\#(A) \end{pmatrix} [0]$$

$$\vec{t}^{(k+1)} = (M)^{T} \vec{t}^{(k)} [0]$$

$$r(a_i) = \lim_{k \to \infty} \vec{t}_i^{(k)} \cdot \#(A) \qquad (EigenTrust)$$

After about 10 iterations this value is sufficiently approximated. Figure 6 shows the development of the reputation in the *EigenTrust*-system.

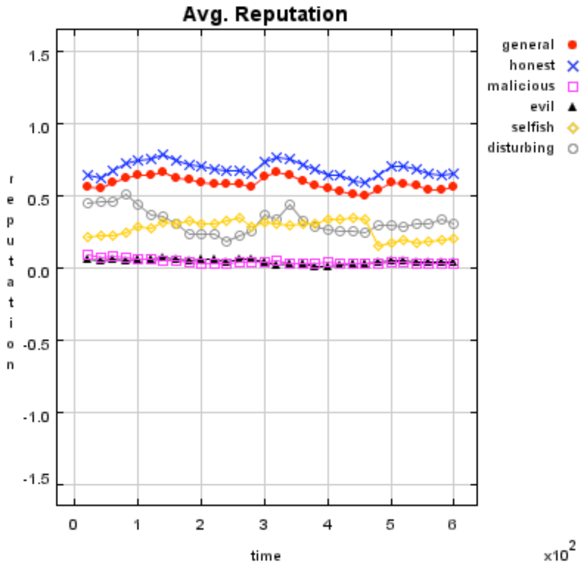


Figure 6: The *EigenTrust*-system

Adaptive Systems

3.14

Within adaptive systems the reputation of an agent changes differently when receiving a new rating, depending on his current reputation. I.e. the reputation of the ratee increases more, if he receives a positive rating and his reputation is low, and changes less if his reputation is already high. Reputation can also decrease in the same manner.

3.15

The Sporas-system (Zacharia and Maes 2000) grants only positive ratings: $\rho : A \times E \rightarrow \{0.1, \ldots, 1\}$. This causes that reentering the community under a new identity is useless, as the reputation will be lower then the current reputation. The resulting value of a given rating depends on the reputation of the rater, too. The higher the rater's reputation is, the larger is the value of his rating. The reputation of an agent $a \in A$ having received a rating from b at time $t \in T$ computes with

$$N_i(a) = r(a, i-1)/D$$

$$\Phi(r(a, i-1)) = 1 - \frac{1}{1+\exp\left(\frac{-(R(a,i-1)-D)}{\sigma}\right)} [0]$$

$$r(a, 0) = 1$$

$$r(a, t) = r(a, t-1) + \frac{1}{\Theta} \cdot \Phi(r(a, t-1))$$

$$\cdot\, r(b, t) \cdot (\rho(a, e_{a,b}^{t}) - N_t(a)) \qquad (Sporas)$$

at which D is the maximum reachable reputation and Θ and σ are constants for the time- and reputation-dependable weight. In the simulation we used D = 30, Θ = 10, and σ = 0.8.

3.16

A similar system providing positive and negative ratings and reputations is suggested by Yu and Singh (2000). The possible ratings are $\rho : A \times E \rightarrow \{\alpha, \beta\}$, where $\alpha > 0$, $\beta < 0$, $|\alpha| < |\beta|$, thus it is easier to drop a good reputation than to build it up. We adapted their proposal to a global approach. The reputation of an agent $a \in A$ at time $t \in T$ computes with

$$\Phi(r, q) = \begin{cases} \alpha & \text{if } r = 0 \wedge q > 0, \\ \beta & \text{if } r = 0 \wedge q < 0, \\ r + \alpha(1 - r) & \text{if } r > 0 \wedge q > 0, \\ \frac{r+\beta}{1-\min\{|r|,|\beta|\}} & \text{if } r > 0 \wedge q < 0, \\ \frac{r+\alpha}{1-\min\{|r|,|\alpha|\}} & \text{if } r < 0 \wedge q > 0, \\ r + \beta(1 + r) & \text{if } r < 0 \wedge q < 0. \end{cases}$$

$$r(a, 0) = 0$$

$$r(a, t) = \Phi(r(a, t - 1), \rho(a, \overline{E_a}[i])) \qquad \text{(YuSingh)}$$

In the simulation we set $\alpha = 0.05$ and $\beta = -0.3$. Figure 7 shows the development of the reputation in the *YuSingh*-system.

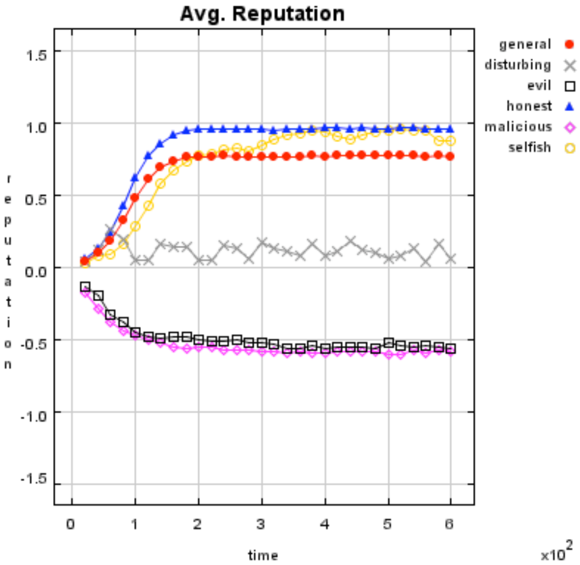


Figure 7: The *YuSingh*-system

Beta Systems

3.17

The *Beta*-System (Jøsang and Ismail 2002) tries to predict statistically an agent's behavior in his next transaction. Therefor the data about earlier transactions is evaluated, and the probability with which an agent behaves good or bad is derived. The share of good (r) and bad (s) transactions an agent made in the past (e.g. neutral behavior means $r = 0.5$ and $s = 0.5$) is determined. These two variables are parameters for the beta-distribution, whose expectation predicts the future behavior of the agent.

3.18

The beta-distribution is a continuous distribution Beta(a, b) between 0 and 1, with two parameters $a, b > 0$ and its expectation $\mathbb{E}$ is $a/(a + b)$. If $a = b = 1$ the beta-distribution is identical to the uniform distribution. If $a > b$ the expectation is > 0.5, if $a < b$ the expectation is < 0.5. The larger a and b are, the smaller is the variance $\sigma^2 = ab/(a + b + 1)(a + b)^2$.

3.19

In this system the ratings $\rho : A \times E \rightarrow \{-1,..., 1\}$ are available. The reputation for an agent a in the

$$r^a = \sum_{i=1}^{\#(\overline{E_a})} \lambda^{\#(\overline{E_a})-i} \cdot (1 + \rho(a, \overline{E_a}[i]))/2$$

$$s^a = \sum_{i=1}^{\#(\overline{E_a})} \lambda^{\#(\overline{E_a})-i} \cdot (1 - \rho(a, \overline{E_a}[i]))/2$$

$$r(a) = \frac{r^a - s^a}{r^a + s^a + 2} \qquad (Beta)$$

where $0 \leq \lambda \leq 1$. When computing the reputation with consideration of the transaction value, the computation for r^a and s^a changes:

$$r^a = \sum_{i=1}^{\#(\overline{E_a})} \lambda^{\#(\overline{E_a})-i} \cdot (1 + \rho(a, \overline{E_a}[i]))/2 \cdot \nu(\overline{E_a}[i])$$

$$s^a = \sum_{i=1}^{\#(\overline{E_a})} \lambda^{\#(\overline{E_a})-i} \cdot (1 - \rho(a, \overline{E_a}[i]))/2 \cdot \nu(\overline{E_a}[i])$$

We simulated this system with λ = 0.99. Figure 8 shows the development of the reputation in the *Beta*-system.

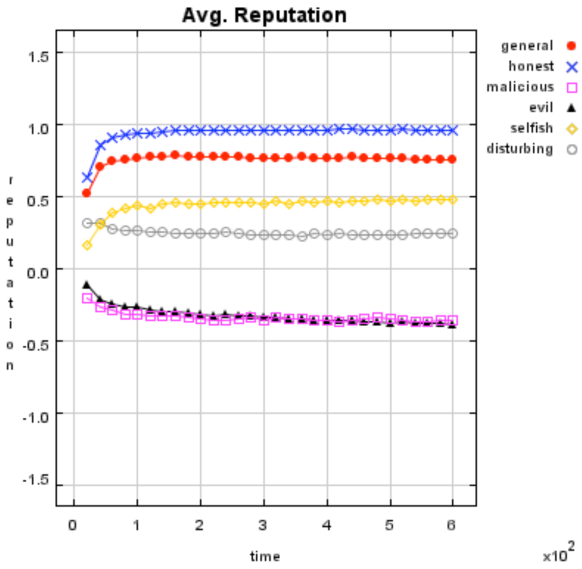


Figure 8: The *Beta*-system

Simulation of Reputation Systems

4.1

There is no accepted test-bed scenario for reputation systems that allows to compare different metrics under the same conditions. Often the *Prisoner's Dilemma* or a custom scenario is used for experimental evaluation of a single system. Sabater (2002) proposes a scenario inspired by a supply chain with a number of different markets that allows quite complex settings. In our work the simple scenario of a space where agents provide homogeneous symmetric services to each other is preferred. We have evaluated the efficiency of the different metrics by simulation of this scenario. As a simulation framework we used Repast. RePast is an agent-based simulation toolkit which offers easy to use methods for visualization. The simulation is based on discrete time ticks. At each tick every agent is supposed to do something, in our case to trade with another agent and rate him. After the agents finished their actions the data is collected and visualized.

4.2

After a short description of the framework's capabilities, we present the different agent models we implemented. Then we evaluate how the different metrics given in section 3 performed against several attacks by these agents.

Framework

4.3

Our simulation framework is highly automated. The handling of the agents, the initiation of transaction, and the storage of the ratings are part of the framework. The only thing that must be implemented for simulating a new metric $\mathcal{M}$ are the functions ρ and r.

4.4

The steps of a transaction are depicted in figure 9. When the simulation engine selects an agent to initiate a transaction with another agent, he first tests if the transaction is acceptable. It is also checked if the other agent is willing to transact with him. The acceptance function is described in section 4.9. Then the initiating agent determines the outcome of the transaction and both agents submit their ratings. These ratings depend on the agent type and may not match the real outcome. The different types of agents we used in our simulation framework are described in section 4.12. New agent types can be easily integrated by implementing a template class.

4.5

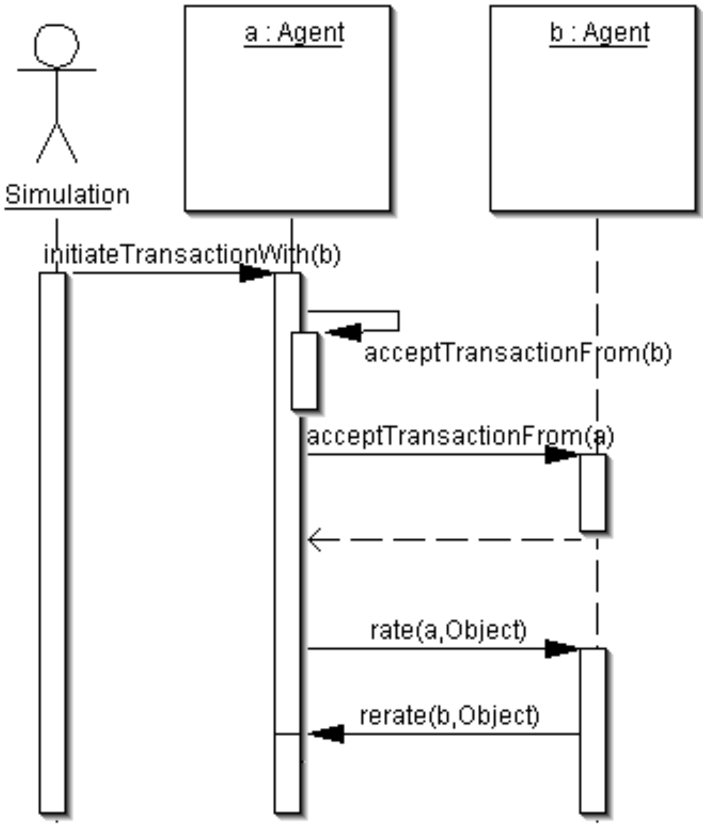


Figure 9: The rating process

4.6

For the correct interpretation of the outcome of encounters, the framework needs a mapping of the possible outcomes of transactions to the set of possible ratings: $\tilde{\rho} : [-1,\ldots,1] \rightarrow Q$. This function is used by the rating function ρ such that the agent can rate his partner corresponding to his intention. The worst rating is $\tilde{\rho}(-1)$, the best one is $\tilde{\rho}(1)$. For example the *eBay*-system we implemented uses this function:

$$\tilde{\rho}(o) = \begin{cases} -1 & \text{if } o < -0.2, \\ 1 & \text{if } o > 0.2, \\ 0 & \text{else.} \end{cases}$$

4.7

The metric function *r* is evaluated by the framework after each step and can be used by the agents to retrieve information about other peers. It computes the current reputation of $a \in A$.

The storage object and the agent *a* of interest are passed as arguments. Methods for retrieving all necessary information, i.e. the received ratings $\rho(a, e)$ and the context information $\tau(e)$ and $\nu(e)$ for each $e \in E_a$, are provided by the storage object. To influence the environment of the simulation, our framework provides many user-definable parameters.

Agent Models

4.8

The effectiveness of a reputation system and its metric depends on its resistance against several types of attacks. This section describes the different simulated agent behaviors. The success of non-honest agents is the measurement for the quality of the metric. The different agent types implemented for the simulation are called *honest*, *malicious*, *evil*, *selfish*, and *disturbing*. These types differ in their behavior when transacting and rating. But they have in common, that their decision whether they are willing to transact with another peer or not, is based on the reputation of this peer. Thus we first present our model for acceptance behavior and after that the differences between the agent types.

Acceptance Behavior

4.9

We assume that no agent will transact voluntarily with a non-trustworthy agent. Thus the peer's reputation has high influence on an agent's decision if he should transact or not. The higher the reputation of the transaction partner is the more likely he wants to trade with him. Another aspect which influences the agent's decision is the value of the transaction. When trading lower values an agent should not expect such a high reputation as he would when trading high value goods. The method which is described below models the behavior we expected the agents to have. The usability of this method is proofed empirically by our simulation runs. This method works better than other methods with simple tresholds like in Buchegger and Le Boudec (2003) without considering transaction values. However, this mechanism is still far away from modeling a realistic human behavior.

4.10

We use a scale-based approach. For every reputation system, a reputation $s > 0$ must be known, at which an agent is expected to be 100% trustworthy. The other way around we expect an agent with the reputation $-s$ to be totally untrustworthy. Based on this we divide the reputation space in scale segments á $\frac{1}{10}s$. Based on an empirical study we define the values from table 1 for the value of *s*. The scale segment

$$\left[\frac{r_s-1}{10}s, \frac{r_s}{10}s\right), r_s \in \mathbb{N}$$

has the number r_s, and for negative reputations the segment

$$\left[\frac{r_s}{10}s, \frac{r_s+1}{10}s\right), -r_s \in \mathbb{N}$$

has the number r_s. During the simulation the agents trade goods with the value $v \in [1,..., 100]$. These values are divided in five equal segments, too. They are numbered like the reputation scale. The value v lies in the segment $v_s = \lceil 1/20v \rceil$. For instance the value $v = 25$ falls in the segment 2.

Table 1: Acceptance reputation

System	**Reputation s**
eBay	120
Simple	100
SimpleValue	2000
Value	2000
Average	1
AverageSimple	1
AverageSimpleValue	40
AverageValue	40
Blurred	6
BlurredValue	250
OnlyLast	0.8
OnlyLastValue	40
EigenTrust	2
Sporas	30
YuSingh	1
Beta	0.8
BetaValue	0.8

4.11 Every agent $a \in A$ is classified by his reputation r and the transaction value v. The classification c_s is computed with $r_s(a) - v_s(a)$, where $r_s(a)$ computes the segment on the reputation scale for the agent a and $v_s(a)$ the segment on the value scale. Figure 10 shows the computation of the classification. The classification c_s is used as a parameter of the beta-distribution. We chose this distribution, because it is easily configurable by the two parameters to the desired distribution. If $c_s > 0$ the parameters for the beta-distribution are $a = c_s$ and $b = 1$, if $c_s < 0$ it is a = 1 and b = $- c_S$. Now every agent decides with a treshold $t \in [0,..., 1]$ if he wants to accept the transaction, t < Beta(a, b), or deny it, t > Beta(a, b). The higher the treshold t is, the less risk an agent is willing to agree to.

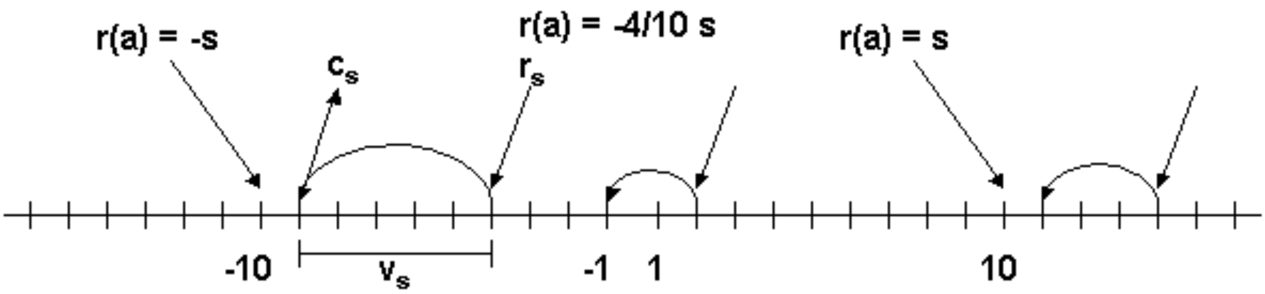


Figure 10: Computation of the classification

4.12

Reputation systems which have only positive reputations (cf. section 3.12 *EigenTrust*), can also be handled by this model. The weakness of such systems is, that a newbie and an agent with bad reputation do not differ. Thus it is acceptable, that malicious agents are treated like agents with neutral reputation by this model. In the simulation of *EigenTrust* in (Kamvar, Schlosser, and Garcia-Molina 2003) agents accepted transactions from other peers with the (worst) reputation 0 with the fixed probability of 10%.

Agent Types

4.13

The agents types we used for simulation differ in their rating-behavior and the sort of transactions they initiate. It was not our aim to model a realistic behavior or to include elaborate strategies, but to develop some simple test cases for our framework. The different types are:

Honest Agent This agent initiates only good transactions (i.e. he serves his trade partner what he expected he would get). His ratings are always correct (good transactions are rated good, and bad transactions are rated bad).

Malicious Agent This agent initiates good, neutral and bad transactions by chance. He tries to undermine the system with his rating behavior and rates every transaction negative.

Evil Agent or Conspirative Agent These agents try to gather a high reputation by building a group in which they know each other. If an evil agent finds another evil agent to trade with, they always give each other a good rating. If an evil agent does not find another evil agent, after seeking for a while, he transact neutral and rates neutral.

Selfish Agent This agent is a so called *freerider*. He blocks all inquiries by other agents and refuses to rate his transaction partners. He just initiates neutral to good transactions by himself.

Disturbing Agent This agent tries to build a high reputation, such that the other agents trust him, with making good transactions and correct rating. Then he switches to a malicious behavior until his reputation gets too bad and then starts from the beginning.

4.14

Example If an honest agent h initiates a transaction with a malicious agent m , the outcome of this transaction is positive, thus h tells m that he should be rated with $\vec{\rho}(1)$. But m rates h with $\vec{\rho}(- 1)$, that is why we call him malicious.

4.15

The naming of these agent types is quite arbitrary and should just provide an intuitive meaning of their behavior. They may not be confound with different agent types used in other papers, as there is no consistent naming policy for different agent types in the context of reputation systems known to us.

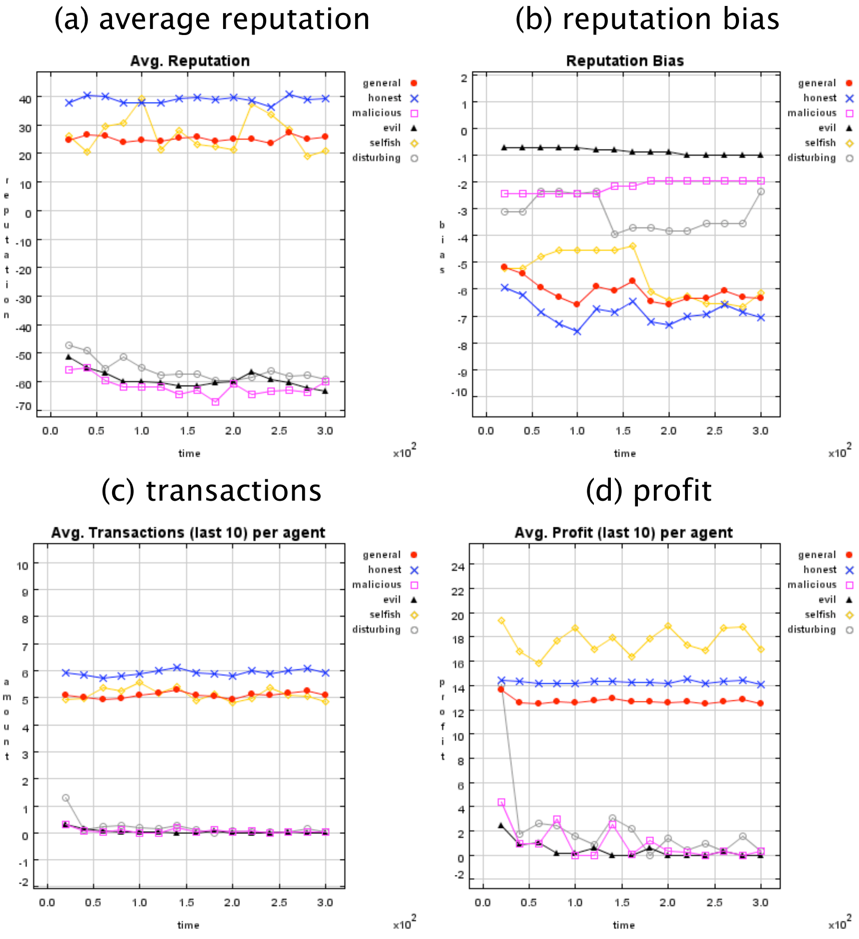
Evaluation Criteria

4.16

First we have to define the criteria which determine if a metric works well. In the simulation we measure the average reputation, the reputation bias, the transaction rate, and the profit separately for each agent type.

4.17

The *average reputation* is the mean value of all reputations of agents from a specific type. We store the correct rating an agent should receive for each transaction. The difference between the average reputation which is calculated with these stored ratings and the actual ratings is the *reputation bias*. The *transaction rate* is the portion of transactions completed successfully compared with all initiated transactions. The *profit p* of a transaction calculates by $p = \frac{1}{o+2} v$ where *v* is the transaction value and *o* $\in$ [− 1,..., 1] is the real outcome of this transaction. The assumption is, that the profit is higher, if an agent cheats on another agent. Figure 11 illustrates an example of the measured values. The horizontal axis is the time axis in all cases, as usual. In figures 11(a) and 11(b) the vertical axis represents the computed reputation respectively the bias of this reputation. Figure 11(c) shows the amount of successfully finished transactions from the last 10 initiated ones, figure 11(d) has an abstract scale on the vertical axis, this values have to be interpreted relatively to each other.


(a) average reputation

(b) reputation bias

(c) transactions

(d) profit

Figure 11: Evaluation Criteria

4.18

The only agent types with concrete aims are the *evil* and *disturbing* agents. The evil agents want to raise their reputation and the disturbing agents try to gain high profit. The *malicious* and *selfish* agents do not receive any advantages from their behavior. Thus for these attacks it is only important, if the *honest* agents retain their good reputations.

Simulation Results

4.19

We simulated the different metrics in several simulation runs with an increasing amount of hostile agents in each run to find the critical point, where the metric fails to maintain dependable reputation. During each simulation run the distribution of agents did not change fundamentally. Agents may enter or leave the community, but since this happens equitable likely, the initial distribution will only vary slightly. It was not the aim to simulate a real world behavior but to use our framework under extreme conditions. This should give an example how it can be used to compare different metrics. Table 2 summarizes the results of the simulation. The values depicted there are the maximum amount of hostile agents at which the reputation systems is able to provide a dependable reputation. All other agent type amounts were uniformly distributed initially.

4.20

The *AverageSimple*-system is not listed there, because it turned out to be totally impractical. Even honest agents could not reach high reputation, after a while all agents had neutral reputation. The reason for this may be that the negative ratings from the malicious agents are weighed too much.

Table 2: Strengths & weaknesses of the metrics

System	Dist.	Evil	Mal.	Self.
eBay	–	50	60	+
Simple	–	30	70	+
SimpleValue	–	30	70	+
Value	–	50	60	+
Average	–	60	70	+
AverageSimpleValue	–	60	70	+
AverageValue	–	60	60	+
Blurred	–	50	70	+
Blurred-Value	–	50	70	+
OnlyLast	+	+	50	+
OnlyLastValue	+	+	30	+
EigenTrust	–	+	60	30
Sporas	–	30	70	60
YuSingh	–	50	50	60
Beta	–	50	70	+
BetaValue	–	60	70	+

+	:	resistent up to 80% of this type
x	:	at an amount of *x*% of this type, the system fails
–	:	the system does not protect against this attack

4.21

A first surprising result is the strength of the *OnlyLast*-system. This is the only system which can resist an attack from disturbing agents, and can also stand evil and selfish agents. This corroborates the theoretically derived results from Dellarocas (2003). Dellarocas proved that the efficiency of an accumulative system (he uses the term binary feedback system) is independent of the number of ratings *n* summarized by the mechanism. According to his results an *OnlyLast*-

system (which is a binary system with n = 1) is more efficient in environments where agents can change their identities without cost. On the other hand it is the weakest systems against malicious agents. The reason for this effect is, that it is very hard for an honest agent, to trade with others and receive a good rating, if he already has a (false) negative reputation, because a malicious agent rated him negative.

4.22

The *EigenTrust*-system is the only system except from the *OnlyLast*-systems which is resistent to attacks from evil agents. But this system is very susceptible to attacks from selfish agents. The reason is that this system depends on groups of agents who trust each other. If only few agents rate other agents, the system fails to recognize groups and cannot compute dependable reputations.

4.23

Against attacks from malicious agents as well as selfish agents the most systems are equally good. Just the *OnlyLast* and *YuSingh*-systems cannot handle malicious agents, and the *EigenTrust*-, *Sporas*, and *YuSingh*-systems fail on large amounts of selfish agents. The weakest system in this simulation was the *Sporas*-system which supports only positive ratings and reputations. A possible reason for this is our model for the acceptance behavior. Additionally, it is hard to determine reasonably if the same positive reputation is good or bad without information e.g. about the amount of encounters an agent had.

4.24

With this results we tried to combine different metrics to compensate for the individual weaknesses. After some experiments with modifications of the *BetaValue*-system we focused on the *Average*- and the *OnlyLast*-system. Both systems can be understood as summing up the previous ratings of an agent using different weights. In the *Average* system all ratings have the same weight, in the *OnlyLast*-system the recent rating has the weight 1 and all other ratings have the weight 0. The *Blurred*-system is somewhere in between, but could not handle the disturbing agents. Thus we decided to interpolate between the *Average* and the *OnlyLast*-system by weighting the ratings not linear, like we did in the *Blurred*-system, but quadratic, so that the recent ratings have more influence on the reputation. The resulting metric $\mathscr{M} = (\rho, r)$ is:

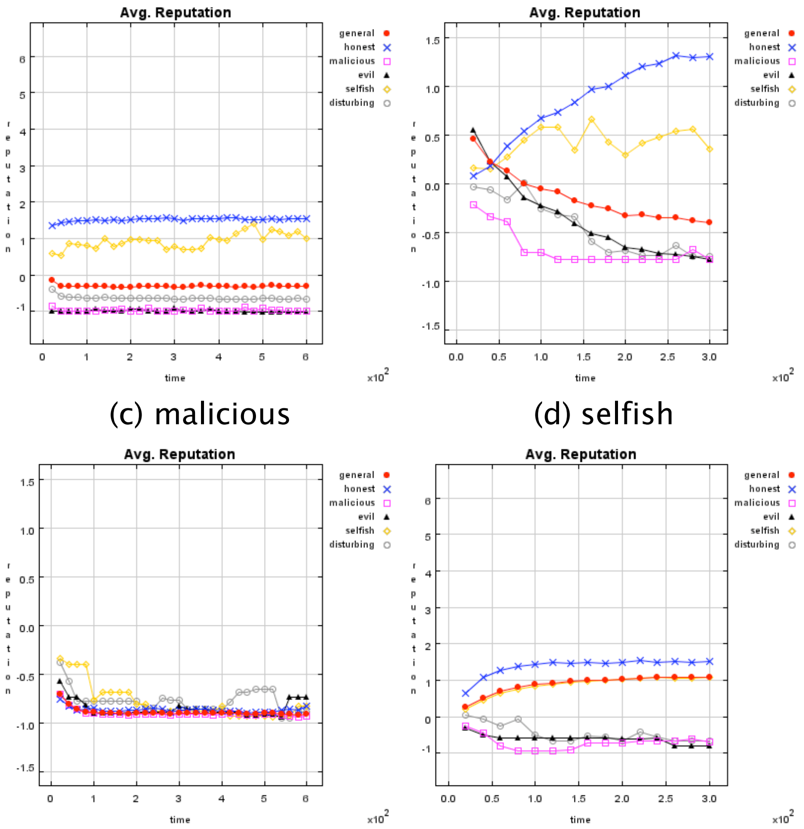
$$\rho : A \times E \rightarrow \{-1, 0, 1\}$$

$$r(a) = \sum_{i=1}^{\#(\overline{E_a})} \frac{\rho(a,\overline{E_a}[i])}{(\#(\overline{E_a})-i+1)^2}$$

4.25

We call this system *BlurredSquared*. This system is invulnerable to disturbing, evil, and selfish agents. It resists malicious agent up to an amount of 60%. The average reputation at the maximum amount of hostile agents is illustrated in figure 12. In figure 12(a) 80% of all agents are disturbing agents. Similarly, in figures 12(b), 12(c), and 12(d) we have the maximum amount of evil, malicious, and selfish agents. To us the resistance against disturbing agents is far more important than the resistance against an unnatural high amount of malicious agents. Thus with this new system we attenuated the weakness of the *OnlyLast* system against malicious agents a little.

(a) disturbing (b) evil



(c) malicious

(d) selfish

Figure 12: The *BlurredSquared*-system

4.26

The amounts of hostile agents we are looking at in our simulation seem unnaturally high on first sight. But in the simulation these hostile agents spread their behavior equally over the whole community, while in reality they will focus on certain agents. Thus the amounts we found here would be in the same relation in reality, but much lower, if the aim of hostile agents were to harm individual agents and not the whole community.

Related Work

5.1

Most papers that propose a new reputation system evaluate its efficiency by simulation, but compare it only with one single system or a scenario without any reputation system. There is no work known to the authors that tries to provide an overall comparison of a number of reputation systems. Dellarocas (2003) has provided some theoretical results that are applicable to accumulative systems. Mui has developed a typology of reputation system and developed a model of trust and reputation. However, his understanding of the term reputation is different to ours. Sabater (2002) proposes a framework called *SuppWorld* as a test-bed scenario for reputation systems. It is not known if this framework has been used for other purposes than to test the ReGret system. Fullam et al. (2005) recently proposed a new testbed for trust and reputation mechanisms. The idea of this testbed is not to test a single reputation system in different communities, but to provide a framework in which different agents using different trust building mechanisms shall compete. We are curious about the acceptance of this testbed in the community.

Conclusion

6.1

We presented a formal model which can be used to describe reputation systems, especially their metric. We gave an overview of the different kinds of metrics in global reputation systems and used our model to describe them. Furthermore we gave an approach how a trust decision can be computed based on reputation, the scale-based acceptance behavior, and introduced models for the major types of agents related to reputation systems. Based on the model and the agent types we implemented a generic framework for reputation metrics that allowed us to compare arbitrary metrics. By simulation we found the strengths and weaknesses of the different metrics and proposed the *BlurredSquared* metric.

6.2

There is no formal mechanism yet to prove the suitability of a reputation metric for a given scenario. This motivates the need for a broadly accepted testbed. We hope that our framework is useful for others to provide a well-defined base for the test of new metrics.

6.3

In the future, we plan to extend our model and the simulation framework to include support for local reputation systems and more complex contexts. We also see the need for more sophisticated types of agents and a better understanding of the acceptance function. Finally, a set of standard agent communities that model real-world application domains (e.g. auctions) would be required to enable more specific experiments. We will make the code, a description and additional simulation results available from our web page (ITO 2004).

References

S. Buchegger and J.-Y. Le Boudec. (2003) A Robust Reputation System for Mobile Ad-hoc Networks. *Technical Report, EPFL, Switzerland*.

J. Carbo, J. M. Molina, and J. Davila. (2003) Trust management through Fuzzy Reputation. In *Int. Journal of Cooperative Information Systems*, 12(1): 135-55.

E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. (2002) A Reputation-based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In *Proc. of the 9th ACM Conference on Computer and Communications Security*, Washington, DC, USA, November, pp. 207-16.

C. Dellarocas. (2000) Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior. In *ACM Conference on Electronic Commerce*, pp. 150-7.

C. Dellarocas. (2003) Efficiency and Robustness of eBay-like Online Feedback Mechanisms in Environments with Moral Hazard. *Working paper, Sloan School of Management, MIT, Cambridge, MA*, January.

eBay Homepage. (2004) http://www.ebay.com.

E. Friedman and P. Resnick. (2001) The Social Cost of Cheap Pseudonyms. *Journal of Economics and Management Strategy*, 10(2): 173-99.

K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. Rosenschein, L. Vercouter, M. Voss. (2005) A Specification of the Agent Reputation and Trust (A R T) Testbed: Experimentation and Competition for Trust in Agent Societies. In *Autonomous Agents and Multi Agent Systems (AAMAS 05)*, July, pp. 512-8.

T. D. Huynh, N. R. Jennings, and N. Shadbolt. (2004) Developing an Integrated Trust and Reputation Model for Open Multi-Agent Systems. In *Autonomous Agents and Multi Agent Systems (AAMAS 04), Workshop on Trust in Agent Societies*, July, pp. 65-74.

ITO. Project Reputation Homepage. (2004) http://www.ito.tu-darmstadt.de/projects/reputation.

A. Jøsang and R. Ismail. (2002) The Beta Reputation System. In *Proc. of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia, 17-19 June 2002*, pp. 324-37.

R. Jurca and B. Faltings. (2003) Towards Incentive-Compatible Reputation Management. In *Trust, Reputation and Security, AAMAS 2002 International Workshop*, volume 2631 of *Lecture Notes in Computer Science*. Spinger, pp. 138-47.

S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. (2003) The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the twelfth international conference on*

World Wide Web. ACM Press, pp. 640-51.

G. Mahoney. (2002) Trust, Distributed Systems, and the Sybil Attack. PANDA Seminar Talk, University of Victoria, Canada.

E. M. Maximilien and M. P. Singh. (2003) An Ontology for Web Service Ratings and Reputations. In S. Cranefield, T. W. Finin, V. A. M. Tamma, and S. Willmott, editors, *Proceedings of the Workshop on Ontologies in Agent Systems (OAS 2003) at the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia, July 15, 2003*, volume 73 of *CEUR Workshop Proceedings*, pp. 25-30.

L. Mui, M. Mohtashemi, C. Ang, P. Szolovits, and A. Halberstadt. (2001) Ratings in Distributed Systems: A Bayesian Approach,. In *Proceedings of the Workshop on Information Technologies and Systems (WITS'2001)*.

L. Mui, M. Mohtashemi, and A. Halberstadt. (2002) Notions of Reputation in Multi-Agents Systems: a Review. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, ACM Press, pp. 280-7.

RePast Homepage. http://repast.sourceforge.net.

P. Resnick and R. Zeckhauser. (2000) Reputation Systems. *Communications of the ACM*, 43: 45-8.

P. Resnick and R. Zeckhauser. (2002) Trust Among Strangers in Internet Transactions: Empirical Analysis of ebay's Reputation System. The Economics of the Internet and E-Commerce. *Advances in Applied Microeconomics*, 11: 127-57.

J. Sabater. (2002) *Trust and reputation for agent societies*. PhD thesis, Universitat Autonoma de Barcelona.

J. Sabater and C. Sierra. (2001) REGRET: A Reputation Model for Gregarious Societies. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 194-5.

J. Sabater and C. Sierra. (2002) Reputation and Social Network Analysis in Multi-Agent Systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 475-82.

A. A. Selçuk, E. Uzun, and M. R. Pariente. (2004) A Reputation-based Trust Management System for P2P Networks. In *Proceedings of the 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid04)*.

A. Whitby, A. Jøsang, and J. Indulska. (2004) Filtering out Unfair Ratings in Bayesian Reputation Systems. In *Autonomous Agents and Multi Agent Systems (AAMAS 04), Workshop on Trust in Agent Societies*, July.

L. Xiong and L. Liu. (2004) PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities. In *IEEE Transactions on Knowledge and Data Engineering (TKDE), Special Issue on Peer-to-Peer Based Data Management*, pp. 843-57.

B. Yu and M. P. Singh. (2000) A Social Mechanism of Reputation Management in Electronic Communities. In *Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace*, pp. 154-65.

G. Zacharia and P. Maes. (2000) Trust Management through Reputation Mechanisms. *Applied Artificial Intelligence*, 14: 881-907.

Return to Contents of this issue