A New View on Normativeness in Distributed Reputation Systems

Beyond Behavioral Beliefs[1]

Philipp Obreiter[1] and Birgitta König-Ries[2]

[1] Institute for Program Structures and Data Organization
Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
obreiter@ipd.uni-karlsruhe.de
[2] Institute of Computer Science
Friedrich-Schiller-Universität Jena, 07743 Jena, Germany
koenig@informatik.uni-jena.de

Abstract. Autonomous entities in artificial societies are only willing to cooperate with entities they trust. Reputation systems keep track of the entities' behavior and, thus, are a widely used means to support trust formation. In a P2P network, the reputation system needs to be distributed to the individual entities. In previous work, we have shown that some of the limitations of distributed reputation systems can be overcome by making use of hard evidence. In this paper, we take this idea one step further by deriving beliefs of others' trustworthiness from one's own experiences and the available hard evidence. For this purpose, we justify why a self-interested autonomous entity may choose to behave according to the norms of the system designer. As a consequence, the proposed belief model does not only incorporate behavioral beliefs but also beliefs regarding the normativeness of an entity. We prescribe how beliefs are revised if new evidence becomes available. The introduced models for recommendations and belief formation enable us to prove that self-interested entities always issue truthful recommendations regarding transactional behavior. The simulative evaluation shows that a self-interested entity can be expected to be normative and, thus, to comply with our system design.

1 Introduction

If you look at computer systems, there is a clear trend away from closed monolithic systems towards self-organizing artificial societies composed of autonomous entities with no central control and no commonly trusted unit. Examples are peer-to-peer systems, open multi-agent systems, and ad hoc networks. All these systems have a number of characteristics in common: In order to achieve their individual goal, it is necessary for the entities in the system to cooperate. However, due to their autonomy, on the one hand, entities will only cooperate, if it is beneficial to them, on the other hand, entities

[1] The work done for this paper is funded by the German Research Community (DFG) in the context of the priority program (SPP) no. 1140. The authors would like to thank Michael Klein, Jens Nimis and Sokshee Goh for their comments on this paper. In addition, we are grateful for Peter Reiher's comments on the legal obstacles for tampering software.

are able to cheat in the course of a cooperation. In order to avoid being cheated on, an entity will only cooperate with entities it trusts.

Distributed reputation systems are a commonly suggested means to support trust formation [1–4]. They allow for the exchange of information about certain entities' behavior and make it thus possible to base trusting decisions not only on one's own prior experience with that entity but also on others' experiences. The major challenge for the design of distributed reputation systems consists of accurately estimating others' behavior based on the information at hands. In previous work [5], we have argued that some of the exchanged information should be non-repudiable (and, thus, become *hard evidence*) in order to improve the accuracy of the estimation. Other authors [4] propose the inclusion of *norms* for the same purpose. However, existing distributed reputation systems cannot make use of the additional information provided by hard evidence and norms. For the most part, this is due to their ignorance of non-repudiability and their fixation on behavioral information. In this paper, we make up for these deficiencies by redesigning distributed reputation systems. We mainly contribute **(1)** by justifying the consideration of norms for systems of self-interested autonomous entities, and **(2)** by providing a multi-layered *belief model* that derives the belief of others' normativeness from own experiences and hard evidence.

The remainder of this paper is structured as follows: In Section 2, we extend the basic system model by considering hard evidence and norms. In Section 3, we show that existing approaches fail to exploit relevant information for the formation of beliefs. Based on this analysis, we make up for this deficiency by proposing a novel belief model in Section 4. We evaluate the properties of the redesigned distributed reputation system in Section 5 and, finally, conclude the paper in Section 6.

2 System Model

In this section, we present the system model that is assumed for the remainder of this paper. For this purpose, we describe the basic system model and extend it in two directions: **(1)** Based on the ideas of our previous work [5], non-repudiability is proposed as a means of acquiring hard evidence. In this context, a recommendation model based on hard evidence is described. **(2)** We suggest that system design should make use of norms and provide a justification of this idea. The justification is valid even for systems that consist of self-interested autonomous entities.

2.1 Basic System Model

The system consists of *entities* that may enter into *transactions* at any time. Each transaction occurs between a pair of entities (*transaction peers*). Each transaction peer executes an *action* on behalf of the transaction partner[2] who is able to check whether the action has been executed correctly. The autonomy of the entities implies that an entity may *defect* by failing to execute its action. Take for example two entities of a

[2] This assumption of mutually beneficial transactions may be relaxed by making use of non-repudiable promises [6].

P2P network that agree on exchanging a pair of documents. After having received the document of the transaction partner, a transaction peer may defect by refusing to transmit its promised document. The *reputation system* keeps track of defections in order to caution the entities about the defectors. In the absence of a central component, the reputation system is distributed to the entities themselves. More specifically, each entity runs a *local instance* of the reputation system. As a prerequisite for the operation of a distributed reputation system, the entities have to be able to send *authenticated messages*. This means that the recipient of a message knows which entity has sent it. For this purpose, each entity has a unique and unalterable *identity*. The local instances of the reputation system may cooperate by exchanging *recommendations*. The issuer of a recommendation (*recommender*) communicates information regarding a certain entity (*recommendee*) to the *recipient* of the recommendation.

2.2 Evidential Extension of the System Model: Non-repudiability and Hard Evidence

In our previous work [5], we have pointed out that distributed reputation systems should make use of non-repudiability. In the following, we recapitulate this idea and extend the system model accordingly.

Each entity is able to issue *non-repudiable* tokens and verify the validity of the non-repudiable tokens that have been issued by others. By this means, the issuer of such a token is able to non-repudiably commit to a statement. The token itself provides *hard evidence* of this commitment. Hence, we refer to non-repudiable tokens as hard evidence in the remainder of this paper. As we have pointed out in [5], presuming a means of non-repudiability is practically not a stronger assumption than presuming a means of authenticating messages.

According to the basic system model, a transaction consists of the exchange of repudiable actions between the transaction peers. This transactional model is extended as follows: The transaction peers exchange a pair of non-repudiable tokens before and after the proper transaction. First, each transaction peer commits to the imminent transaction and its terms by issuing a *contract* to its transaction partner. After the exchange of actions, each transaction peer issues a *receipt* and, thus, confirms that its transaction partner has complied with the transaction terms as promised. In the absence of a trusted third party, defections are still possible by retaining one's own contract, action or receipt.

The availability of contracts and receipts leads to a redefinition of the recommendation model. It dispenses with (potentially fake) reports of beliefs. Instead of that, a recommendation consists of hard evidence and is required to be non-repudiable. By this means, a recommender is forced to commit to the content of his recommendations. We distinguish between three types of recommendations: **(1)** *Disrecommendations:* The perception of cooperation is documented by a receipt. In a disrecommendation, an entity commits to such a perception. A disrecommendation is required to be non-repudiable so that its recipient has hard evidence of such commitment. Furthermore, we apply the policy that, in order to be valid, a disrecommendation has to enclose the contract of the entity that reportedly defected [5]. By this means, disrecommendations are always based on transactions that actually took place. **(2)** *Self-recommendations:* Each

entity may disseminate its receipts by issuing self-recommendations [7]. **(3)** *Inconsistency proofs:* An inconsistency proof can be furnished if an entity issues non-repudiable commitments that are mutually incompatible. This is the case if an entity issues both a receipt and a disrecommendation regarding the same transaction.

2.3 Normative Extension of the System Model: Justification and Norm Design

In the past, it has been proposed to include norms into the design of distributed reputation systems [4]. However, the authors fail to justify why a self-interested autonomous entity could possibly decide to abide with norms that are detrimental to itself. In the following, we provide such a justification. Furthermore, we provide a thorough discussion of how norms should be designed. In addition, we define the type of an entity based on its normativeness.

System design and autonomy. According to the basic system model, each entity is *autonomous* and, thus, cannot be forced to behave in a certain way. Instead of that, each entity is only controlled by its human principal. For example, in a P2P system like KaZaA, a piece of software constitutes the entity and the user owning the hardware represents the human principal. If the user is not pleased with the performance of the software, he can remove or tamper it. The user does not have to be an expert for doing so if he has access to a tampered version of the software that meets his demands.

What are the consequences if each entity can be arbitrarily tampered? The system designer conceives a set of algorithms that should be run by the participants of the system. Traditionally, it is argued, that, if any part of the algorithms is not *incentive compatible*, the designer has to expect that the entities are tampered. Therefore, incentive compatibility of any behavior becomes the key criterion of system design (e.g., [8]). However, this is not completely true.

Tampering costs and compliance costs. In the following, we argue that – contrary to popular belief – tampering does incur some costs (*tampering costs*) and that, as a consequence, system design is disburdened of some difficulties.

A human principal may tamper his entity either by *creating* a tampered version of the software or by *adopting* the tampered version of others. Both options violate laws for a couple of reasons: **(1)** Tampering includes re-engineering of the software. In the US, this is explicitly forbidden by the Digital Millennium Copyright Act if the software is protected by a technical means [9]. **(2)** A tempered version represents a derived work. Hence, its creation or distribution infringes copyright law. This also applies the adoption of a tampered version since it incurs downloading and, thus, duplicating the derived work [10]. **(3)** Contractual law is violated, too, if the system designer protects his software by an adequate licence. In contrary to the US, such contractual protection is forbidden in the EU by the Software Directive §6.1 [11]. Still, the system designer could demand that the users agree with a licence regarding the identities that he assigns to them. According to that additional licence, identities may only be used in connection with the original software. By this means, the use of tampered versions infringes contractual law even in the EU.

Furthermore, there are tampering costs that are specific to the creation or adoption of tampered versions: The creation of a tampered version requires expert skills and is rendered even more costly by the means of code obfuscation [12]. On the other hand, the adoption of a tampered version exposes the user to risks due to the intransparency of its behavior. It could perform worse than the original version or even be a trojan. Consequently, tampering one's own entity always incurs costs.

We do not claim that these costs are prohibitive. Rather, we argue that system design should make use of these tampering costs, even if they are small. This means that system design could foresee some behavior that is not fully incentive compatible. As a result, complying with the system design also incurs some costs (*compliance costs*). It is clear that entities are tampered whenever these compliance costs exceed the tampering costs. For this purpose, system design has to keep the compliance costs as marginal as possible. This rules out systems in which the participants are designed to behave altruistically[3]. However, it makes sense to design a system in which proposed behavior may be not fully incentive compatible under infrequent circumstances.

Norm design. A norm refers to a non-enforceable rule given by the system designer (i.e., r-norms [13]). We propose to incorporate *norms* into the design of distributed reputation systems. The presence of norms leads us to defining the *type* of entities based on their normativeness: An entity's type is *normative* if the entity always complies with the norms. Hence, normative entities adhere to the original system software. Contrarily, its type is *strategic* if it decides whether to comply depending on the circumstances. Therefore, strategic entities run a tampered version of the system software. In the following, we discuss which norms should be included into the design of the system.

Norm design has to reconcile two conflicting demands. **(D1)** Norms should prescribe cooperative behavior in order to allow a population of norm compliant entities to perform well. **(D2)** Norms have to be self-enforcing by rendering norm compliance incentive compatible. If norms are not sufficiently self-enforcing, the compliance costs surpass tampering costs so that entities are tampered and deviate from the norm. In order to obtain self-enforcing norms, we propose to orientate norm design towards two maxims. **(D2a)** The only means of being perceived as *normative* entity is to actually abide with the norms. For this purpose, behavior that is prescribed in a norm has to be highly perceptible by others. **(D2b)** Each entity wants to be perceived as normative entity, i.e., as an entity that always complies with the norms. The maxims create a momentum towards self-interested norm compliance.

We propose two norms: **(N1)** *Never defect* in a transaction after having agreed on participating in it. **(N2)** *Never* issue *inconsistent* statement about the same issue. These norms meet the above demands of norm design: **(D1)** Both norms prescribe cooperative behavior. **(D2a)** Compliance with norm (N1) is perceptible to the transaction partner. Furthermore, compliance with norm (N2) has the potential of being fully perceptible by any entity if statements are required to be non-repudiable. **(D2b)** Since a transaction represents a win-win situation, each entity desires to participate in as much transactions as possible. For the choice of transaction partners, an entity prefers those entities that

[3] For example, the reputation mechanism of KaZaA has been hacked because it presumes that each entity truthfully calculates and disseminates its reputation.

are least likely to defect. Normative entities abide with norm (N1). Hence, entities want to be perceived as normative in order to be preferred as transaction partners. By an analogous argumentation for norm (N2), we obtain that each entity wants to be perceived as normative so that its statements are given more weight.

3 Exploiting Information for the Formation of Beliefs

In the previous section, the system model has been extended in order to account for hard evidence and norms. In this section, we point out that a distributed reputation system should exploit additional information that arises from these extensions. Furthermore, we discuss behavioral beliefs and show that their formation is the ultimate goal of a distributed reputation system. Based on these preconsiderations, we review existing distributed reputation systems.

3.1 Information and Behavioral Beliefs

An entity runs a local instance of the distributed reputation system in order to obtain support for its trusting decisions. In the following, we take a closer look at the general set-up for the provision of such support. The treatment is divided into two steps. First, we examine which type of information is available as input to a local instance of the reputation system. Second, we show that behavioral beliefs are required as output in order to support trusting decisions.

Information. The most obvious source of information are first-hand experiences. In the course of a transaction, the transaction partner may cooperate or defect. Since norm (N1) prescribes "never defect", these two cases correspond to normative and non-normative behavior respectively. In the following, we denote normative behavior by entity Y with $N_Y^{(b)}$ and non-normative (strategic) behavior with $S_Y^{(b)}$. Therefore, first hand experiences regarding entity Y consist of a sequence of $N_Y^{(b)}$ and $S_Y^{(b)}$. An entity has to consider the first hand experiences made by others that are communicated in recommendations. The recommendation model of Section 2.2 ensures that the contents of the recommendation relates to transactions and conflicts that actually occurred. Furthermore, transactional behavior is context-dependent [14]. This means that, even if an entity always behaves well in a specific context (e.g., low value transactions), it could still misbehave in other contexts. Hence, an entity should make use of *context information* in order to assess transactional behavior.

Behavioral beliefs. The decision whether to participate in a transaction represents a trusting decision. In order to make this decision, an entity has to predict the likely behavior of the potential transaction partner (say Y). Such a prediction has to cope with two types of uncertainty [15]: *Aleatory uncertainty* (or stochastic uncertainty) results from the fact that the transaction partner may behave in random ways. This means that there exists an intrinsic probability $p(N_Y^{(b)})$ that Y behaves cooperatively in the forthcoming transaction. In contrast, *epistemic uncertainty* (or subjective uncertainty) ensues from the lack of knowledge about the transaction partner. Therefore, the probability has

to be estimated according to one's own current beliefs [16]. We denote such subjective estimate by entity X with $p_X(N_Y^{(b)})$. Since the estimate is based on X's beliefs and regards Y's behavior, we refer to it as *behavioral belief* of X regarding Y.

We elaborate on three important issues of behavioral beliefs. First, a behavioral belief is a probabilistic belief due to the aleatory uncertainty. Second, a behavioral belief is fallible due to the epistemic uncertainty. Therefore, it might be necessary to revise it if new information becomes available. Third, the probabilistic interpretation of behavioral beliefs provides for a straightforward means of making trusting decisions. More specifically, an entity decides to participate in a transaction if its expected utility is positive [2].

3.2 From Information to Behavioral Beliefs: Existing Approaches

In the following, we analyze how existing approaches of distributed reputation systems derive behavioral beliefs from the information at hand. We focus our analysis in two directions: **(1)** We do not consider approaches that make use of a central component in order to manage reputation (e.g., [17]) or foresee side-payments (e.g., [8]). **(2)** An approach is not taken into account if it does not provide for probabilistic estimations of behavior. This is because such estimations are a prerequisite for utilitarian decision making. Examples of approaches that fail to fulfill this requirement are [3].

The approaches of [1, 2] presume that the inert probability $p(N_Y^{(b)})$ of cooperative behavior by Y is the same for each transaction of Y. In such a case, Y's behavior follows a Bernoulli distribution of $N_Y^{(b)}$ and $S_Y^{(b)}$. Based on this assumption, the beta function is proposed as probability density function regarding $p(N_Y^{(b)})$ [1]. By this means, both aleatory and epistemic uncertainty are taken into account. Furthermore, first hand experiences can be directly integrated into the parameters of the beta function. However, this approach lacks a theoretically founded means of integrating others' first-hand experiences. Therefore, the maximum likelihood estimation of $p(N_Y^{(b)})$ is suggested in [2]. This provides a straightforward means of integrating others' first-hand experiences.

We argue that behavioral approaches suffer from three deficiencies: **(1)** The consideration of others' first-hand experiences is solely based on plausibility and dispenses with hard evidence. According to [5], this yields several limitations. **(2)** The approaches do not allow for the integration of type information. Even if entity Y was known to be normative, the probability $p(N_Y^{(b)})$ is not necessarily 1 since the entity could defect unintendedly. The other way round, a strategic entity Y does not have to defect in every transaction. **(3)** The approaches are based on the assumption that the inert probability of cooperative behavior is the same for each transaction. This inhibits the use of context information. Therefore, it has been proposed to provide separate behavioral beliefs for a set of potentially interrelated context categories [18]. Yet, the definition of the categories' granularity is difficult because it has to trade off the imprecision of aggregating contexts with the overhead of managing several separate behavioral beliefs for each entity.

4 The Multi-Layered Belief Model and Belief Revision

Existing approaches apply too narrow models of beliefs that cannot exploit the information at hands. Therefore, in this section, we redesign the belief model by proposing several novel concepts: **(1)** Type beliefs are modelled such that epistemic uncertainty is taken into account. **(2)** Beliefs regarding type and behavior are interrelated by a multi-layered mapping. It explicitly models context-dependent norm abidance and unintended defection. **(3)** The revision strategy of type beliefs is able to take any relevant information (including behavioral information) into account.

4.1 The Belief Model

Apart from behavioral beliefs, we propose to make use of beliefs regarding an entity's type and intentions. In addition, we interrelate beliefs by suggesting the mappings type-to-intention and intention-to-behavior. The ensuing three layers of beliefs are illustrated in Figure 1.

Type beliefs and intention beliefs. In order to capture epistemic uncertainty, we model a belief regarding an entity's type as a probabilistic belief. For this purpose, we introduce some further notation: We denote the fact that entity Y is normative/strategic with $N_Y^{(t)}$ and $S_Y^{(t)}$ respectively (the superscript (t) refers to Y's type). Thus, the *type belief* of entity X regarding entity Y is the subjective probability $p_X(N_Y^{(t)})$. According to Section 2.3, an entity is either normative or strategic. Hence, there is no aleatory uncertainty about an entity's type. Consequently, a type belief may be expressed as a simple probability. Contrarily, the existing behavioral approaches have to make use of probability density functions in order to account for both aleatory and epistemic uncertainty.

In order to interrelate type beliefs and behavioral beliefs, we introduce an intermediate kind of belief regarding intentions. Entity Y may intend to abide with norm (N1) by refraining from defection. We denote this fact with $N_Y^{(i)}$. If Y intends to break the norm by defecting, its intention is strategic (denoted with $S_Y^{(i)}$). Thus, an *intention belief*[4] of X regarding Y is the subjective probability $p_X(N_Y^{(i)})$.

Type-to-intention mapping. An entity's intention is derived from its type. Normative entities always intend to abide with the norms, hence $p_X(N_Y^{(i)}|N_Y^{(t)}) = 1$. However, strategic entities abide with the norms only if they want to. Their decision of norm abidance is based on the context γ of the transaction. In the following, we denote the subjective probability $p_X(N_Y^{(i)}|S_Y^{(t)},\gamma)$ that a strategic entity intends norm abidance with $p_X^{(n)}(\gamma)$. If the transaction value v is the main driving force of context-dependent behavior, a simple estimate of this probability is $e^{-\kappa v}$ with some positive parameter κ. This type-to-intention mapping incorporates context-dependence more seamlessly than the existing behavioral approaches. This is because it solves their conflict between imprecision of aggregating contexts and overhead of separate context categories.

[4] This definition is compatible to the BDI-architecture [19]. It refers to X's beliefs regarding Y's intention.

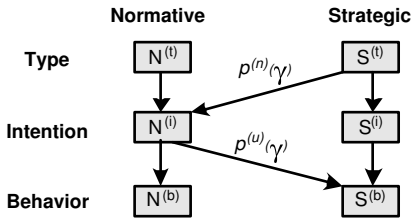


Fig. 1. Derivation of behavioral beliefs from type beliefs

Intention-to-behavior mapping. An entity's behavior is derived from its intention. The intention to abide with the norms is a prerequisite for actually abiding with them, hence $p_X(N_Y^{(b)}|S_Y^{(i)},\gamma) = 0$. However, norms can be broken unintendedly. The probability of unintended defection depends on the context of the transaction. For instance, partitioning is more likely to occur in long running transactions. Therefore, we have to estimate the conditional probability $p_X(S_Y^{(b)}|N_Y^{(i)},\gamma)$ that we denote with $p_X^{(u)}(\gamma)$ in the following.

The estimation of unintended defection is considerably easier than the one of strategic norm abidance. This is because unintended defection is due to the transaction's environment (i.e., nature [20]) that behaves non-strategically. Hence, it suffices to be able to estimate the probability of partitioning and node failure in order to estimate the subjective probability $p_X^{(u)}(\gamma)$ appropriately. In such a case, this probability only contains aleatory uncertainty.

4.2 The Belief Revision

The belief state of an entity consists of its type beliefs regarding the entities it is acquainted with. Whenever previously unknown information becomes available, type beliefs have to be revised. In the following, we provide a probabilistically sound means of such belief revision.

Let us assume that entity X perceives *cooperation* or *defection* of its transaction partner Y for the transaction context γ. In such a case, X's type belief regarding Y is revised according to Bayes' formula. The required conditional probabilities are derived from the formulas of Section 4.1. For perceived cooperation ($N_Y^{(b)}$), belief revision is as follows[5]:

$$p_X(N_Y^{(t)}|N_Y^{(b)},\gamma) = \frac{p_X(N_Y^{(t)}) \cdot p_X(N_Y^{(b)}|N_Y^{(t)},\gamma)}{p_X(N_Y^{(b)}|\gamma)} \tag{1}$$

Disrecommendations are considered as follows: If entity Y disrecommends entity Z, we have to presume that Y or Z defected, i.e., $S_{YZ}^{(b)}$ occurred. We cannot infer that Z defected since Y could have disrecommended after having defected itself. Upon receival of such a disrecommendation, we revise the beliefs regarding Z based on Bayes'

[5] The formula of perceived defection is obtained by replacing $N_Y^{(b)}$ with $S_Y^{(b)}$.

formula:

$$p_X(N_Z^{(t)}|S_{YZ}^{(b)},\gamma) = \frac{p_X(N_Z^{(t)}) \cdot p_X(S_{YZ}^{(b)}|N_Z^{(t)},\gamma)}{p_X(S_{YZ}^{(b)},\gamma)} \quad (2)$$

The belief regarding Y cannot be revised because, otherwise, the issuance of disrecommendations is not incentive compatible any more. Hence, the belief regarding Y is only revised if it is disrecommended by Z. This is no restriction since Z possesses the contract that is needed for issuing a disrecommendation.

Belief revision due to a *self-recommendation* leads to rehabilitation: A prior disrecommendation could be refuted by the receipt that is enclosed in the self-recommendation. In such a case, an inconsistency proof regarding the disrecommender Y becomes available. Before updating the beliefs regarding Y, the disrecommendee (and self-recommender) Z has to be rehabilitated from its downgrading of equation (2). This is done by inverting the equation based on the present type belief regarding Y.

An *inconsistency proof* evinces that an entity is tampered. If such proof regarding entity Y is available, the entity is believed to be certainly strategic, i.e., $p_X(N_Y^{(t)}) = 0$.

5 Evaluation

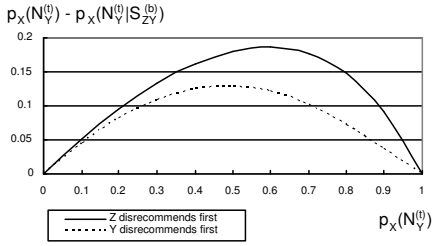
The previous sections have shown how a distributed reputation system has to be redesigned in order to account for hard evidence and norms. In this section, we discuss and evaluate two issues that ensue from the redesign: **(1)** We analyze under which circumstances strategic entities disrecommend. The analysis shows that defective behavior always leads to disrecommendations. **(2)** We determine simulatively the costs that normative entities have to bear in order to comply with the norms. By this means, we evaluate to which degree the system's norms are self-enforcing.

5.1 The Disrecommendation Game

The motivation of discussing disrecommendation behavior is twofold. On the one hand, most existing approaches fail to provide a means for rendering the issuance of disrecommendations individually rational. We show that our system provides for such a means and, thus, guarantees that defective behavior of strategic entities is widely perceived as such. On the other hand, the analysis clarifies the relationship between the two novel concepts of our approach, i.e., the belief model and the recommendation model.

The analysis of disrecommendation behavior is based on the following situation. Let us assume that, during a transaction between entity Y and Z, a transaction peer defected by failing to execute its action. In such a case, both entities possess a contract but lack a receipt of the transaction and, thus, are able to mutually disrecommend. However, a strategic Y or Z only chooses to disrecommend if it is in its interests to do so. In the following, we provide a game-theoretic analysis of this situation.

According to Section 2.2 and 4.2, Y is only able to disrecommend Z to a third party X if X requested from Y such a disrecommendation. Upon receival of this disrecommendation, X revises its type belief $p_X(N_Z^{(t)})$ regarding Z. More specifically, the belief



		Z	
		no DR	DR
Y	no DR	$(0,0)$	$(-c_d,-c_i)$
	DR	$(-c_i,-c_d)$	$(-c_i-c_d+\frac{b_f}{2},-c_i-c_d+\frac{b_f}{2})$

Fig. 2. (a) Impact of disrecommending first and **(b)** the Disrecommendation Game

regarding Z's normativeness is degraded since it could be the originator of the defection. As a result of the degradation, X becomes less willing to transact with Z. Hence, the fact of being disrecommended incurs some costs c_d for Z. On the other hand, Y has to bear the costs c_i of issuing the non-repudiable disrecommendation and handing it over to X. Consequently, the disrecommendation appears to be detrimental for both the Y and Z. However, the prescription of belief revision provides a counterweight for the costs of disrecommending. If Y knew that Z subsequently disrecommends it to X, it could preemptively decrease X's type belief regarding Z by disrecommending Z first. By this means, the impact of Z's disrecommendation is decreased. Figure 2(a) illustrates[6] these considerations. It interrelates X's prior type belief regarding Y (*x-axis*) with the relative degradation of the type belief after having considered Z's disrecommendation (*y-axis*). The impact of Z's disrecommendation is considerably lower if Y disrecommends Z first. We conclude that being the first to disrecommend provides for a comparative benefit b_f. For virtually every application area of our system model (e.g., P2P systems), the synergies of inter-entity transactions outweigh by far the overhead of issuing a non-repudiable token. Therefore, we presume that the comparative benefit b_f of disrecommending first largely exceeds the costs c_i of issuing a disrecommendation.

We summarize these considerations in the *disrecommendation game*. Its normal form is shown in Figure 2(b). If both Y and Z choose to disrecommend (DR), they are equally likely to disrecommend first. Therefore, their expected comparative benefit of disrecommending first is $\frac{b_f}{2}$. If this benefit is higher than the disrecommendation costs c_i, we obtain a coordination game. This means that Y would decide to do as Z if it knew how Z decides and vice versa. More technically speaking [20], there are two stable equilibria in pure strategies, i.e., (DR, DR) and $(\neg DR, \neg DR)$. Furthermore, the equilibrium in mixed strategies consists of Y and Z disrecommending with the probability $\frac{2c_i}{b_f}$. This equilibrium is unstable since, whenever Y deviates from this equilibrium strategy by increasing (decreasing) the probability of disrecommending, Z decides to always (never) disrecommend.

The derivation of equilibrium strategies is based on the assumption that Y and Z behave rationally. This is the case if both Y and Z are strategic entities. However, according to the prescription of Section 2.2, normative entities always disrecommend. This raises the question how a strategic Y would decide depending on its type belief

[6] The illustration is based on $p^{(n)}(\gamma) = 30\%$, $p^{(u)}(\gamma) = 5\%$ and the prior belief $p_X(N_Z^{(t)}) = 50\%$.

$p_Y(N_Z^{(t)})$ regarding Z. The probability $p_Y(N_Z^{(t)})$ provides a lower bound of the probability that Z decides to disrecommend. Hence, we derive $p_Y(N_Z^{(t)}) > \frac{2c_i}{b_f}$ as a sufficient condition that a strategic Y always disrecommends. Due to $b_f \gg c_i$, this condition is fulfilled for virtually every belief of Y. Consequently, *strategic entities decide to disrecommend under most circumstances*.

The desirable outcome of above analysis is based on the two key concepts of our approach. On the one hand, the *recommendation model* ensures that disrecommendations are only possible for transactions in which a defection actually occurred. By this means, the disrecommendation game is only played by a pair of entities that had a conflict during their transaction. On the other hand, the *belief model* is exploited twice. First, the prescription of *belief revision* yields the comparative benefit of disrecommending first and minimal disrecommendation costs. Second, the solution of the disrecommendation game is based on the presence of *normative* entities that are pre-committed to norm abidance.

5.2 Simulative Quantification of the Compliance Costs

According to Section 2.3, an entity decides to remain normative as long as the costs of complying with the norms do not exceed the costs of tampering the original system software. In the following, we simulatively quantify the compliance costs in order to assess under which circumstances normativeness is a rational choice.

We have implemented our approach in DIANEmu [21]. The most important aspects of the simulation setting are as follows[7]: **(1)** *Benchmark:* The system consists of 100 entities. The number of normative entities varies between 20% and 95%. No a priori knowledge exists among the entities. Each entity obtains between 5 and 25 opportunities to choose its transaction partner among 2 entities. The transaction value is distributed uniformly in $[0.5, 1.5]$. The probability of unintended defection by any peer is 5%. **(2)** *Configuration of normative entities:* The estimation of the conditional probabilities are set by $\kappa = 0.5$ and $p_X^{(u)}(\gamma) = 0.05$. **(3)** *Configuration of strategic entities:* A strategic entity defects if its transaction partner executes its action first. In such cases, the defected transaction partner is disrecommended whenever possible. **(4)** *Metric:* The compliance costs are defined as the difference between the average utility of strategic and normative entities.

Figure 3 shows the simulation results. It appears that the compliance costs tend to decrease for an increasing number of transactions or an increasing ratio of normative entities. We interpret the results by making three quantitative conclusions: **(1)** Irrespective of the setting, the compliance costs never exceed 6, which is the equivalent value of defecting in 6 transactions. Therefore, a human principal runs the original version of the system software if his tampering costs outweigh the benefits of defecting 6 times. **(2)** Irrespective of the number of transactions, the compliance costs become negative for a sufficient high ratio of normative entities. In such cases, an average normative entity outperforms an average strategic entity. Since tampering costs are non-negative,

[7] Due to space limitations, we describe the setting in detail in a technical appendix. It is available at *http://www.ipd.uka.de/~obreiter/iTrust05techApp.pdf*.

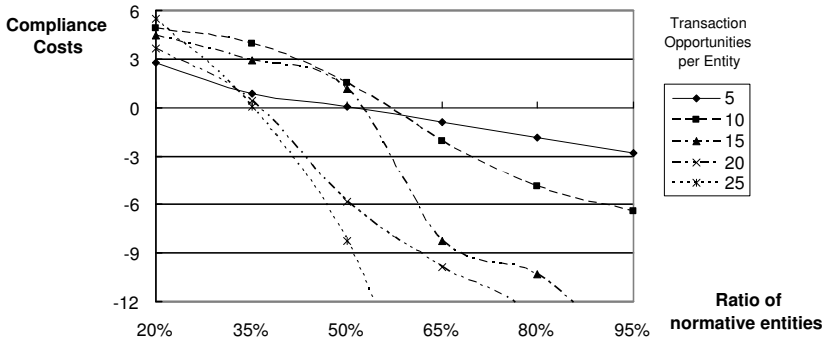


Fig. 3. The costs of complying with the norms

the system becomes completely normative if the ratio of normative entities exceeds a certain threshold (between 35% and 60% depending on the number of transactions). **(3)** Based on the first two points, we are able to interpret the overall system's dynamics: A fully normative system is in an equilibrium state. The equilibrium is very stable since norms are self-enforcing unless 40% of the entities (or even 80% for tampering costs beyond 6) are irrationally tampered.

6 Conclusion

Distributed reputation systems provide a means for restricting misbehavior in self-organizing systems of autonomous entities. Previous work has suggested the inclusion of hard evidence and norms into distributed reputation systems. In this paper, we have justified why system design should make use of norms. The presence of norms has led us to the distinction of *normative* and *strategic* entities. We have shown that existing distributed reputation systems cannot make use of the additional information provided by hard evidence and norms. We have made up for these deficiencies by redesigning distributed reputation systems. For the integration of hard evidence, we have suggested a novel *recommendation model* that is built on three types of recommendations. Furthermore, we have provided a multi-layered *belief model* that incorporates type beliefs. By this means, we are able to capture different types of informational input. We have considered in detail the mapping of type beliefs to behavioral beliefs and the revision of type beliefs based on behavioral information. The analysis of the disrecommendation game has shown that all entities issue disrecommendations regarding transactional behavior whenever they are able to do so. Finally, we have demonstrated simulatively that cooperative behavior is self-enforcing if the ratio of normative entities is at least moderate.

In the future, we aim at integrating a means of bailing for another entity's normativeness [7]. In this context, we will investigate how the availability of hard evidence regarding such bails influences belief formation and self-recommendations. Furthermore, we plan to compare simulatively the properties of our system with other existing distributed reputation systems.

References

1. Buchegger, S., Boudec, J.Y.L.: A robust reputation system for P2P and mobile ad-hoc networks. In: Second Workshop on the Economics of Peer-to-Peer Systems. (2004)
2. Despotovic, Z., Aberer, K.: A probabilistic approach to predict peers' performance in P2P networks. In: 8th Intl Workshop on Cooperative Information Agents (CIA'04). (2004)
3. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in P2P networks. In: WWW2003. (2003)
4. Castelfranchi, C., Conte, R., Paolucci, M.: Normative reputation and the costs of compliance. Journal of Artificial Societies and Social Simulation **1** (1998)
5. Obreiter, P.: A case for evidence-aware distributed reputation systems. In: Second International Conference on Trust Management (iTrust'04), Oxford, UK, Springer LNCS 2995 (2004) 33–47
6. Obreiter, P., Nimis, J.: A taxonomy of incentive patterns - the design space of incentives for cooperation. In: Second Intl. Workshop on Agents and Peer-to-Peer Computing (AP2PC'03), Springer LNCS 2872, Melbourne, Australia (2003)
7. Obreiter, P., Fähnrich, S., Nimis, J.: How social structure improves distributed reputation systems - three hypotheses. In: Third Intl. Workshop on Agents and Peer-to-Peer Computing (AP2PC'04), To appear in post-proceedings, New York (2004)
8. Jurca, R., Faltings, B.: Towards incentive-compatible reputation management. In et al., R.F., ed.: AAMAS'02-Workshop on Deception, Fraud and Trust in Agent Societies, Springer LNAI 2631 (2003)
9. Jones, P.: Software, reverse engineering and the law (2005) http://lwn.net/Articles/134642/.
10. Hoffman, I.: Derivative works (2002) http://www.ivanhoffman.com/derivative2.html.
11. Council of the European Communities: Software directive – council directive on the legal protection of computer programs (91/250/EEC) (1991)
12. Linn, C., Debray, S.: Obfuscation of executable code to improve resistance to static disassembly. In: Proceedings of the 10th ACM Conference on Computer and Communication Security. (2003) 290–299
13. Tuomela, R.: The Importance of Us: A Philosophical Study of Basic Social Norms. Stanford University Press, Stanford, California (1995)
14. Mui, L., Halberstadt, A., Mohtashemi, M.: Notions of reputation in multi-agents systems: A review. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02), Bologna, Italy (2002)
15. Helton, J.: Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. Journal of Statistical Computation and Simulation **57** (1997) 3–76
16. Bacchus, F.: Probabilistic belief logics. In: Proceedings of European Conference on Artificial Intelligence (ECAI-90). (1990) 59–64
17. Josang, A., Ismail, R.: The beta reputation system. In: 15th Bled Conference on Electronic Commerce, Bled, Slovenia (2002)
18. Kinateder, M., Rothermel, K.: Architecture and algorithms for a distributed reputation system. In Nixon, P., Terzis, S., eds.: Proc. Of the First Intl. Conf. On Trust Management (iTrust), Heraklion, Greece, Springer LNCS 2692 (2003) 1–16
19. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In Allen, J., Fikes, R., Sandewall, E., eds.: 2nd Intl Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, CA, USA (1991) 473–484
20. Rasmusen, E.: Games and Information : An Introduction to Game Theory. Oxford Blackwell (1989)
21. Klein, M.: DIANEmu – a java-based generic simulation environment for distributed protocols. Technical Report 2003-7, Universität Karlsruhe, Faculty of Informatics (2003)