Behavioural Evaluation of Reputation-Based Trust Systems

Sini Ruohomaa, Lea Kutvonen

To cite this version:

Sini Ruohomaa, Lea Kutvonen. Behavioural Evaluation of Reputation-Based Trust Systems. 5th International Working Conference on Enterprise Interoperability (IWEI), Mar 2013, Enschede, Netherlands. pp.158-171, 10.1007/978-3-642-36796-0_14 . hal-01474208

HAL Id: hal-01474208
https://hal.inria.fr/hal-01474208
Submitted on 22 Feb 2017

Behavioural Evaluation of Reputation-Based Trust Systems

Sini Ruohomaa and Lea Kutvonen

University of Helsinki, Department of Computer Science,
P.O. Box 68, FI-00014 University of Helsinki, Finland
{sini.ruohomaa,lea.kutvonen}@cs.helsinki.fi
http://cinco.cs.helsinki.fi/

Abstract. In the field of trust and reputation systems research, there is a need for common and more mature evaluation metrics for the purpose of producing meaningful comparisons of system proposals. In the state of the art, evaluations are based on simulated comparisons of how quickly negative reputation reports spread in the network or which decision policy gains more points against others in a specific gamelike setting, for example. We propose a next step in identifying criteria for a maturity model on the behavioural analysis of reputation-based trust systems.

Key words: Trust management, reputation systems, inter-enterprise collaboration, simulation-based benchmarking, attack models

1 Introduction

The goal of this methodological work is to advance the state of the art of evaluating reputation-based trust management systems. We find that the field currently suffers from a confusion of what kind of evidence simulation experiments can provide exactly, and there is a need for credibly evaluating the attack resistance and robustness of proposed systems [1]. We acknowledge that other attributes such as usability [2, 3], viability [4], deployability [5] and adjustability to different business situations [6] require attention as well. Instead of a complete maturity model addressing all these aspects, our focus here is on trying to advance behavioural evaluation of reputation-based trust systems specifically.

We first summarize the problem setting of the field from the point of view of inter-enterprise collaborations, which are the context of our work [3]. Collaborations take place between autonomous business services operating in an open service ecosystem. New previously unknown or little known actors can join the ecosystem, and old ones may leave. In this environment, each actor has different goals, which change over time, and it must protect its own integrity by making decisions on whether it trusts another service enough to collaborate with it.

Trust management is the activity of upkeeping and processing information which trust decisions are based on, and a trust management system is an automation tool for the purpose. A trust decision is made by a trustor, gauging its willingness to engage in a given action with a given trustee, given the risks and incentives involved. The key input to a trust decision is reputation information, which is commonly used to evaluate the subjective probability that the trustee will either behave according to the collaboration contract (cooperate), or break the collaboration contract (defect).

Reputation information is divided into two categories: First-hand experiences are gained from the trustor monitoring the outcomes of actions it has engaged in itself, and are generally considered to be error-free within the limits of observability. External experiences are gained from third-party recommenders based on their own first-hand experiences; these actors may have an incentive to provide incorrect information deliberately or can simply disagree based on having observed different kinds of behaviour.

The aim of evaluating reputation-based trust systems in research is often phrased in terms of quantifying an improvement to existing work. The prevalent approach of evaluating trust and reputation systems relies on using simulations to produce evidence that a given trust or reputation system is able to correctly identify well- and misbehaved actors of specific kinds (e.g. [7]). These simulations are typically based on fixed stereotypical behaviour patterns (e.g. [8]), which falls under the field of reliability rather than security [1].

When scoring policy behaviour, it is tempting to set up a benchmark of measuring “correct” and “incorrect” decisions given specific evidence. Unfortunately, this is an oversimplification that relies on a set of quite fragile assumptions: that reputation information captures reality accurately, service providers act predictably enough to follow stereotypical patterns, and actors in the marketplace, especially the attackers, are not particularly resourceful. None of these assumptions can be said to be true in an ecosystem of inter-enterprise collaboration. This discrepancy causes a real danger that by introducing reputation measures into the market with inadequate analysis of their relevant behaviour we end up inviting rampant reputation fraud, and advance ecosystem deterioration by introducing a metric that does not serve its purpose. Farmer and Glass have analyzed the effects of deployed web reputation systems in the real world [9, ch. 5], while deployability and market acceptance analysis of system proposals also gain increasing attention in the field of security [4, 5].

The main overarching goal of behavioural analysis of policies of any kind is to support policy selection, but this choice reflects the actors’ different goals. There are no objectively correct answers. Summarizing policy behaviour given specific input patterns helps this comparison, even if there is no universal correct behaviour. As a special case, the purpose of a reputation-based trust management system is to detect and deter misbehaviour, so we should learn what its vulnerabilities and other costs are. These cannot be benchmarked by fixed loads, but have to be analyzed per system; from a security perspective, it is obviously not enough to conclude that a system is robust against the most popular attack of last year. Higher-level classifications of attacks may support vulnerability analysis in the form of a checklist.

Our research question is: what kinds of tools can we apply to evaluate whether a reputation-based trust management system fulfills its behavioural requirements, and particularly, what metrics could be organized as a reusable benchmark between systems and how?

Section 2 provides background on reputation-based trust management, how trust management systems are directed by policy, and summarizes our simulation experiments and attack resistance evaluation from earlier work. Section 3 presents the state of the art on evaluation methods in the field. Section 4 discusses the possibilities and limitations of different methods, such as simulation experiments in analyzing trust and reputation systems, and the ways to evaluate attack resistance based on methods adopted from computer security. Section 5 concludes.

2 Studying the Behaviour of Trust Management Systems

To support the discussion on development of evaluation methods, we use our own earlier work on trust management as an illustrative example in Section 2.1. During our simulation work summarized in Section 2.2 we learned the current evaluation methods could benefit from the steps we propose in Section 4.

2.1 Reputation-Based Trust Management

The purpose of a trust management system is to handle routine trust decisions on behalf of a human user and to collect and manage the relevant input needed for them, most notably first-hand and third-party reputation information. Third-party experiences must be evaluated for credibility and incorporated into the local body of reputation information with care, as they may include low-quality or intentionallly fraudulent data. Non-routine decisions, which for example involve high risks or cannot be automatically decided on due to insufficient information, must be forwarded to a human user to decide on. This division is explicitly configured.

In order for a deterministic automation system to adjust to different business situations, we must separate policy from implementation in the system and make the former modifiable during runtime. A sufficiently flexible information model allows the automated rules to handle quite complex contexts, such as a situation where the reputation of a minor actor in the collaboration is not spotless, but the monetary losses of any errors it may make are covered by insurance and the collaboration as a whole needs someone to fulfil the role in order to happen. The establishment of metapolicy which determines when a situation is routine and when it requires human intervention, in turn, will pick out cases that are not suitable to be handled automatically. This improves the trustworthiness of the decision-making system itself [6].

The two main policies of a reputation-based trust management system are the trust decision policy and the reputation update policy. The trust decision policy determines, based on input such as reputation information, whether we are willing to collaborate with an actor or not. The reputation update policy, on the other hand, establishes how to handle new reputation information; among other things, it must determine how much weight information from external sources is given over local observations [6]. A trust decision policy must balance the number of possible partners and requirement for positive evidence, while a reputation update policy must weigh information quality and credibilty against the amount of information that is available to support decision-making.

As reputation influences trust decisions and through that collaboration opportunities, it attracts manipulation attempts on competitors' and one's own reputation. This causes challenges for finding a robust reputation update policy that can still utilize the information available to support trust decisions. Example attacks on reputation systems [10] include undeserved negative feedback, collusions of multiple actors to skew a specific actor's reputation up- or downwards, or an actor stuffing the ballot by creating multiple seemingly independent identities in a Sybil attack [11].

When selecting a reputation update policy to protect the trustor from being mislead by external reputation information, we can roughly divide the trustees into four categories:

- Well-reputed actors recommended as trustworthy by high-credibility sources,
- Promising actors recommended as trustworthy by low-credibility sources, but generally unknown by high-credibility sources,
- Shunned actors warned to be untrustworthy either by high-credibility sources or by unanimous low-credibility sources, and
- Mysterious actors receiving either very few or contradictory recommendations.

While all of these categories are more or less subjective perceptions rather than proof of the trustees' actual behaviour and trustworthiness, a good reputation system should generally promote the well-reputed actors and weed out the shunned actors. The two other classes require more careful balancing.

A very risk-averse trustor will prefer not to collaborate with the mysterious actors, independent of whether they offer better terms of service. Should everyone adopt this approach, though, newcomers will have no chance of proving themselves, targets of defamation cannot clear their name, and the service ecosystem will begin to deteriorate. The promising actors face a problem similar to newcomers in that they have not proven themselves enough, but at least they have some recommendations supporting them. On the other hand, it is also easier for a malicious attacker to appear as one of the promising actors rather than a well-reputed one, or to claim that any negative recommendations about it result from reputation attacks rather than honest feedback.

2.2 Evaluating Reputation-Based Trust Management Systems

When evaluating the behaviour of a reputation-based trust management system, the usual interest is in studying whether a given trust decision or reputation update policy responds to a specific requirement, such as identifying actors that follow a specific type of misbehaviour as misbehaving. For trust decision policies, the usual appropriate reaction is then to not engage in collaboration with the actor, while for reputation update policies, it is to reject the likely fraudulent information.

In earlier work, we have summarized the simulations and analysis of example trust decision policies [6]; below, we summarize a reputation update policy experiment, where we have compared the effects that four reputation update policies have on trust decisions when the trust decision policy remains fixed [3, ch. 6.3]. Both experiments share a similar structure: the policies under scrutiny are applied to a set of different simulated experience streams as the sole input. Some of the streams have been optimized against each policy for the simulated attacker to defect as efficiently as possible.

Our experiments make two contributions [3]: The behaviour of a given decision or reputation update policy is illustrated through exposing it to different representative experience streams and plotting the resulting trust decision score. Second, the limitations of each policy are demonstrated by defining the behaviour of an optimal attacker, and calculating how much it is possible for it to benefit by defecting while it maintains its reputation above the level of positive trust decisions.

A reputation update policy determines both whether a new experience is incorporated into an agent's private reputation information storage, and how much weight it should be given in future decision-making. A key input to this decision is the source-dependent credibility of the experience. The studied reputation update policies have been selected to represent different types of solutions to this choice, and we have visualized how effectively they discriminate against ill-behaved actors.

The baseline policy for comparison is "Accepting", which simply incorporates all experiences independent of their credibility. The "Weighted" policy offsets the impact of dubious experiences by weighing them by their credibility: as we consider source credibility to be represented by a real number $c \in [0, 1]$, instead of incrementing the counter for the matching type of experiences with 1 per experience, this policy would increment it by c instead. The "Fixed-cutoff" policy ignores all experiences below a minimal credibility limit C_1, and the "Variable-cutoff" policy compared the so far amassed external experiences' average credibility C_2 to the new item's source-based credibility c and accepts the experience if $c \geq C_2$. This is to ensure that the trustor is open to new experiences when it has nothing better, but does not dilute its reputation storage by low-quality information when it has access to more credible experiences. The policies in question were selected to be understandable to a projected end user, and to take advantage of different features of the information model of the system in order to illustrate its advantages.

We matched our experience streams to the previously discussed well-reputed actors, promising actors with positive but low-credibility reputations, and mysterious actors who receive contradictory recommendations: positive reports from high-credibility sources, and negative from low-credibility sources. Shunned actors were covered in the first simulation [6, 3]. Additional streams demonstrated optimal attacker behaviours.

Optimal attackers were designed to keep their reputation high enough to always ensure a positive trust decision, and the actions they could choose from were cooperating, faking a positive low-credibility experience to boost their reputation, and defecting. Each action was assigned a cost based on its impact [6]. The agent's task was to maximize its score per action taken [12] against each target policy separately. For example, the attacker defecting with a major negative monetary effect to the trustor would gain the attacker +6 points, a minor negative effect +2 points, generating a low-credibility fake experience would be a 0-cost action independent of whether it implied a major or minor positive experience, and actually cooperating would cost -1 or -3 points depending on whether the effect to the trustor was minor or major positive, respectively.

For example, the optimal attacker could generate fake experiences and then defect with major negative effect against the Accepting and Weighted policies, but it would require more fake experiences per defection against the Weighted policy. Both policies mainly suit environments where the vast majority of information is truthful, and the impact of the occasional error is low; they do not work against quickly mass-produced fake experiences. The Fixed-cutoff policy refused all suspicious experiences, but is left with fewer experiences and will not be able to take advantage of promising actors with low-credibility positive experiences only. The Variable-cutoff policy, in turn, could be circumvented with a large number of low-credibility reports before the first defection. We have discussed prompt reaction to notable changes in behaviour in other work [13], and proposed other extensions to the example policies in the thesis [3, 6.3].

3 State of the Art in Evaluation Metrics for Reputation-Based Trust Systems

A reputation-based trust management system implements the preferences of its user, and as such there is no objective "correct" result that could be validated. To discuss the state of the art in simulation experiments, we present experimentation approaches from two categories: simulating marketplace resistance against attackers following given behaviour patterns, and simulating a single actor's competitiveness in a marketplace. The first category corresponds to mechanism design. It sets all actors to use the same decision policies and measures how well the marketplace as a whole resists different kinds of misbehaviour. The second category represents agent design, pitting different decision policies against each other in the same marketplace. It measures an agent's competitiveness on the marketplace, given an existing mechanism it needs to adjust to.

3.1 Reputation Systems in Electronic Marketplaces

Related work presents simulation experiments on the behaviour of different accumulative and probabilistic reputation systems in an electronic marketplace [8, 14, 15]. In such a marketplace, intelligent agents, which correspond to our service providers, perform pairwise brief transactions of buying and selling goods. The marketplace is given a distribution of agents with different behaviour profiles, and each agent type has a decision policy; typically the reputation update policy is equal between all agents, and all experience information is shared. The simulation then measures for example the average number of transactions taken with a given type of agent (honest, malicious, etc).

The basic behaviour profiles of agents are typically very straightforward, such as "honest agents always carry out transactions honestly and give fair ratings", while "malicious agents act honestly or dishonestly by chance, and always give negative ratings" [14]. More complex behaviour can be tied to the marketplace as a whole; for example, a "spamming" agent can otherwise act honestly, but always rate other agents negatively in order to make itself more attractive in comparison [14], or an agent may be an opportunistic defector, adjusting its behaviour based on whether there is anyone in the marketplace who will transact with it [15]. Schlosser et al. define a behaviour profile for a "disturbing" agent as one who first builds a high reputation with good transactions, and then uses up the reputation so gained by defection [8].

Honest agents all use the same decision algorithm, and if they transact frequently with malicious agents, the reputation system has failed to protect the marketplace. Based on this definition, few reputation systems are resistant to the optimal attacker model — even the "disturbing" behaviour model [8] turns out to be aptly named, when in fact it is nothing more than a model for a selfish agent behaving rationally within the limitations set by the environment.

To be able to give conclusive results, the tools of game theory require strict formal abstraction of the environment and agent behaviour; the core problem then becomes how to formulate a question within this vocabulary so that it is "solvable", while ensuring that the result still gives some useful information about real marketplaces.

One of the aspects left out by this simplification is the social control or deterrence effect of these reputation-based sanctioning mechanisms. In other words, the simulations do not measure how much the reputation system cuts down the expected gains from optimized misbehaviour, although they may show that a specific fixed negative behaviour pattern gains less in one system than another. The reputation system will inevitably be one step behind a rational attacker, so in the *prediction* of attacks our systems inevitably fail; the goal is therefore damage control and reducing the payoff of attacks. It should be noted that reputation loss can only ever deter an actor who plans to remain on the market in the future, so final sanctioning should come from the slower but generally effective judicial system.

Our own simulations have studied how a given agent survives against rational selfish agents. They simplify the interaction with other actors into experience input streams. We then specify policies that drop optimal attacker gains below a certain level to reflect the deterrence effect. The difference between fixed and optimal attackers is that within the same cost model, all attacks will bring equal or less gain than the optimal one. This allows policy comparisons. The challenge is finding a sufficiently realistic cost model.

As further examples of analysis against a given attack type, Margolin and Levine have measured the cost of successfully executing a Sybil attack [16], or the cost of extra "votes" gained through the attack in different schemes, and Srivatsa et al. have aimed to minimize attacker gains from fixed oscillatory behaviour such as the aforementioned "disturbing" agent model [17].

3.2 Competitive Agent Simulations

In competitive agent simulations, agents and policies are pitted against each other in a fixed environment. Each actor aims to maximize its own gains. The format of shared reputation information is fixed, but agents can choose their internal data representation themselves.

The Agent Reputation and Trust (ART) testbed [18] has attracted notable attention, but is no longer maintained. The Trust and Reputation Experimentation and Evaluation Testbed (TREET) [19] is a more recent proposal. It is a more flexible comparison tool, but does not include the yearly competition forum that helped ART attract wider research attention. Convincing the research community to adopt a specific testbed or a benchmark is a nontrivial task, and the differences in domain requirements make this even more difficult.

The ART testbed simulates a marketplace of service providers competing to sell their services [18]. The provided service is art evaluation for a customer: producing a real number as close to the unknown correct answer as possible. There are a number of limitations and costs related to providing the service: the agent can evaluate some art correctly, or get incorrect results and ask for help from others to validate its results. A reputation system is included to support requesting the help of other actors. The number of actors is low, 10-20, so in practice collecting direct experience on all of them is reasonably easy.

The learning agents in the testbed should maximize their own measured gains. The testbed specifies fixed prices for how much customers pay for an evaluation ($100), the cost of asking for an evaluation from another actor ($10), and the cost of asking for a reputation value (a real number between 0 and 1) from another actor ($1) [20]. In addition, the agent can spend an arbitrary amount of money for its own evaluation, with the quality of information depending on the money spent. Teacy et al. provide further analysis of the ART testbed [20].

There are a few factors that limit ART's usability as a benchmark environment. Besides limitations of the information model of the testbed itself [3, 19], the design of the testbed has misdirected attention towards secondary features of the game: the winning strategy focused its effort on determining the most profitable amount of money to invest in generating its own opinion, and in general, very little reputation was exchanged between any of the agents [20]. As noted in the evaluations of ART [20], we cannot conclude that an agent's competitiveness in the simulated marketplace necessarily has anything to do with the policy performing well for a real enterprise operating in a real marketplace.

The benefit of competitive testbeds to fixed, deterministic benchmark scoring is that the evaluation system is adaptive: instead of optimizing policies against a fixed setup, researchers must prepare for tradeoffs in a more uncontrolled environment, which brings in new aspects of realism from the point of view of the system adapting to its environment. Contests attract researcher attention for psychological reasons as well, and the feedback and fame for winning can help motivate adjusting one's work to a given common framework of evaluation. This sets high demands for the evaluation framework, which must iteratively aim for a relevant abstraction of the marketplace.

There are limitations to the rational self-interested agent design approach as well: When agent fitness is observed in isolation, ecosystem-wide benefits of the reputation system, such as altruistic punishment [21] and social pressure to follow contracts [22], can easily become eliminated from the scope of the simulation. While online business is no doubt competitive, a market for inter-enterprise collaboration cannot sustain itself on short-term self-interest alone [22]. This may become a notable blind spot for the metric.

4 Benchmarking Trust Management Systems

Like most measurement at its core, simulation experiments are illustrative. They reflect their setup, first and foremost, and the results require validation even for reasonably objective measures such as raw performance. Fixed simulations do not test the system's resistance against anything else than the chosen specialized behaviour patterns. As the ART testbed competition shows, even pitting algorithms against each other in a tesbed may teach us very little about their relative fitness in the world outside the testbed. Test loads from actual ecosystems, once available, will also be selected illustrative datasets.

The behavioural requirements of a system should consider four key questions: 1) What kind of normal, constructive behaviour is expected in the system, 2) how effectively does the system recover from expected problems that are not calculated attacks, such as temporary malfunctions, 3) are the incentives the system creates in line with its role in the domain, and 4) how effectively does the system detect and deter both direct misbehaviour in the domain, and misbehaviour towards the system itself, such as reputation fraud?

The first two categories can be addressed with fixed-input simulations suitable for automated benchmarking. The latter two measure the success of the system in promoting desired behaviour and weeding out misbehaviour; as both incentives and attacks must assume a rational actor, they are not possible to capture by fixed behaviour patterns.

4.1 Repeatable Simulations with Fixed Loads

Like reputation itself, simulated experience about reputation-based trust management systems is a subjective, simplified tool for comparison which only gains meaning when coupled with a purpose-driven valuation. A fitting purpose for applying the same test case across multiple systems would then be to provide classifications to aid policy comparison. While benchmarks cannot capture notable differences in the information models of different systems, they can be used to summarize policies built on compatible information models.

The first, often inexplicit test done by a simulation is whether the core system is feasible to implement and run. Related to this, benchmark loads can be used to test the *efficiency and scalability* of a system that has non-trivial complexity, in terms of processing, communications and storage load caused by the decision-making and reputation processes. A well-argued mathematical model of the system complexity can be accepted as proof by itself, but a simulation result requires validation, as the implementation and the selection of loads adds a layer of possible measurement error.

If the system is implementable, the main question becomes whether it supports the intended activities of the user. In order to define a valuation of what is expected as normal behaviour, the *domain-specific requirements* must be made explicit. A set of metrics (cf. [5]) allows a categorization, and the domain-specific requirements guide metric selection. Metrics should reflect the goals of the system so that its success in fulfilling them can be evaluated. The subjective goals of a system designer can be very specific, however, while comparison across multiple systems should leave space for different policy adopter preferences within the domain as well. As an example of the importance of explicit assumptions, Kerr and Cohen measured that the reactivity of systems that assume truthful reports is better than of those who evaluate and weigh incoming experiences for credibility [7]; on the other hand, in a typical competitive environment, not being able to resist fraudulent reports would instead be a critical failure that renders the system unusable.

Once a domain model has been established, we can use it to define test patterns of *constructive behaviour*; this requirement is often taken for granted in systems concentrating on foiling a specific attack, which may lead to an unusable system in practice. Examples of interesting behaviour to simulate include how the system treats cooperative service providers with different capabilities for service provision, or how a newcomer with no reputation data entering the system is able to get started. On the level of reputation and recommender credibility, the system should be able to take advantage of the reputation reports of new actors besides the old ones, and serve cooperative reporters, also if their observations genuinely differ from those of the majority. There are no objectively correct solutions even for constructive behaviour: for example the goal of supporting newcomers is often in conflict with the goal of defending against re-entry attacks.

As a reliability test, a set of test patterns can be defined to illustrate *recovery from problems* as well, as long as they can be modelled statistically for benchmarking. Examples include reactivity to relevant changes in behaviour, how a service can recover its reputation after a temporary malfunction causes it to become unreliable for a while, a well-behaved user suffering and recovering from a defamation attack of fraudulent negative reports against it, or even load balancing for a service whose high reputation makes it too attractive to other actors in the ecosystem.[1]

Reputation-related problems can occur on two levels as well: the above examples represent the interaction of service provision and reputation, while on the second level actors' credibility as recommenders can suffer a disruption and need recovery. Like newcomer support, recovery support conflicts somewhat with robustness against malicious actors, but is important as a use case because the system is always designed for its non-malicious users. To be accepted by the market and serve its purpose, it must benefit the well-behaved actors enough to offset their cost of participation; otherwise it will not be used.

4.2 Robustness Analysis

When deploying a system that promotes good behaviour and sanctions misbehaviour, we must analyze its effects on rational actors who can adjust their behaviour to maximize their gains. The measurement system creates incentives that affect the behaviour of both benevolent and rational actors aiming to subvert the system. For example, if the actor with the highest number of positive transaction reports has a higher chance of being selected as a collaboration partner, the system provides an incentive to engage in many small transactions rather than a few large ones. These secondary incentives are not necessarily intentional or desired, but they should be included in the analysis of the system.

In the field of security, attacks and defenses form a continuous reactive loop, where new attacks are met with new defenses. When we analyze reputation as a sanctioning mechanism, the threat of reputation loss should hopefully deter deliberate attacks by making them more costly. The assumption is therefore that attackers aim to maximize their gains and to minimize costs, which renders them suitable for game-theoretic minimax analysis [12].

Rational attacker models should always be optimized against a specific policy setting. We should generally not depend on security through obscurity, so the attacker should have knowledge of the policy in use and its current reputation. It should have a set of reasonable strategies to choose from, with costs and values assigned according to the resources needed and what we want to defend against.

In our attacker model, we allowed optional ways to reach the goal of fraudulently making money off other actors: defection from many small transactions or a few large ones, and boosting reputation through fraudulent sources or by cooperating. We assigned a cost to cooperation, because while in a general market setting collaboration does pay off, we primarily wanted to ensure that defection does not, and selected the measurement accordingly.

To support attacker analysis, high-level *attack classifications* may act as a reusable checklist. Relevant attack categories include misbehaviour in service

[1] Load balancing through reputation is more relevant for e.g. routing services in mobile ad hoc networks than heterogeneous environments where all actors use their own policies. In marketplaces, pricing can be used to balance against overload.

provisioning, deliberate omissions and misreporting, conspiracy with other malicious actors to increase own reputation, conspiracy to decrease a competitor's reputation, coercion, replay and forgery to influence non-malicious actors' reports, and privacy violations against other actors e.g. through traffic analysis. In addition, the checklist can include rational but non-malicious grievances such as freeriding, i.e. not constructively participating in the aspects of the system that do not benefit the actor directly. One vulnerability grouping based on a review of existing systems has been presented in earlier work [10]; for an expansion to a checklist kind of design tool, a tree-structured categorization providing additional levels of detail may provide better usability.

Robustness analysis results should be approached with a similar curious scepticism as research prototypes when it comes to evaluating a system's deployability: rather than providing positivistic evidence of specific desirable attributes of the system, the analysis acts as a feedback-collection step in a design science process. In other words, while not coming up with a vulnerability does not prove that it does not exist, going through the exercise of systematically looking for holes in the design is a valuable step in improving system design itself, and a part of good research practice that leads to more mature systems.

4.3 Methods

A benchmark serves best as a summarizing tool that simplifies comparisons. While system designers cannot use a benchmark load to prove the absence of a vulnerability or the objective superiority of a scheme, deployers may well benefit from more standardized comparison frameworks that provide an overview of the tradeoffs made in any specific systems. Towards this goal, we are also working on a first prototype of a simulation-based comparison tool for reputation update policies in order to identify useful patterns for benchmarking.

A categorization framework would help in better capturing the fact that different policies represent different tradeoffs between partially conflicting goals, and as a result suit different environments and business needs. What the specific needs of a given environment are can only be determined by the actors in it [23]. Focusing too intently on specific behaviour patterns carries the risk of overly technology-centric evaluation of the proposed systems, so a balance must be sought between different methods of collecting feedback on a system.

Our own simulation experiments represent an initial step in more generally summarizing policy behaviour given a specific input, such as identifying policies that produce positive trust decisions for trustees who are only known through low-credibility sources but have only positive experiences within them ("accepts promising actors"). This could be used as a basis to develop a more comprehensive categorization-based evaluation framework in the style of what Stajano et al. have established for evaluating user authentication [5].

For attack resistance, our minimax-based analysis of optimal attackers provides a new angle into this kind of evaluation in comparison to the prevalent methods in the field. We have also summarized how we have applied the method in practice; the analysis demonstrates that making impact information (minor and major positive and negative outcomes) and credibility evaluation available for the automation policies improves the attack resistance of the system [3].

5 Conclusion

We have identified benefits and limitations of the state of the art in simulation-driven experimentation on trust and reputation systems, and gauged the potential of different methods for a set of behaviour-related measurement purposes. The two major directions we identify are building benchmarks for the inter-enterprise collaboration setting, and robustness analysis, which is by nature more specialized for each system and its purpose. General classification tools can help with this analysis as well.

Benchmarks can be applied to simplify comparisons between systems. One notable extension to the idea are competitions within a given system; we believe the potential for this approach has not yet been exhausted in the state of the art, although the task of designing a high-quality marketplace abstraction is quite demanding. Attack resistance analysis, on the other hand, does not seem to lend itself to simulation.

Acknowledgments This research has been performed in the Collaborative and Interoperable Computing (CINCO) group at the University of Helsinki, Department of Computer Science.

References

1. Gollmann, D.: From access control to trust management, and back — a petition. In: Trust Management V; 5th IFIP WG 11.11 International Conference, IFIPTM 2011; Proceedings. Volume 358 of IFIP AICT., Copenhagen, Denmark, Springer (June/July 2011) 1–8
2. Marsh, S., Basu, A., Dwyer, N.: Rendering unto Caesar the things that are Caesar's: Complex trust models and human understanding. In: IFIPTM 2012. Number 374 in IFIP AICT, NIT Surat, India (May 2012) 191–200
3. Ruohomaa, S.: The effect of reputation on trust decisions in inter-enterprise collaborations. PhD thesis, University of Helsinki, Department of Computer Science (May 2012)
4. Zibuschka, J., Roßnagel, H.: On some conjectures in IT security: the case for viable security solution. In: Sicherheit, Schutz und Zuverlässigkeit (SICHERHEIT 2012). Volume P-195 of Lecture Notes in Informatics., Bonn, Germany, Gesellschaft für Informatik (2012)
5. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: IEEE Symposium on Security and Privacy, San Francisco, California, USA (May 2012) 553–567
6. Ruohomaa, S., Kutvonen, L.: Trust and distrust in adaptive inter-enterprise collaboration management. Journal of Theoretical and Applied Electronic Commerce Research **5**(2) (August 2010) 118–136

7. Kerr, R., Cohen, R.: Smart cheaters do prosper: Defeating trust and reputation systems. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009). Volume 2., Budapest, Hungary, ACM (May 2009) 993–1000
8. Schlosser, A., Voss, M., Brückner, L.: On the simulation of global reputation systems. Journal of Artificial Societies and Social Simulation **9**(1) (January 2006)
9. Farmer, F.R., Glass, B.: Building Web Reputation Systems. O'Reilly (2010)
10. Yao, Y., Ruohomaa, S., Xu, F.: Addressing common vulnerabilities of reputation systems for electronic commerce. Journal of Theoretical and Applied Electronic Commerce Research **7**(1) (April 2012) 1–15
11. Douceur, J.R.: The Sybil attack. In: Electronic Proceedings of the 1st International Workshop on Peer-to-Peer systems (IPTPS'02), Cambridge, MA, USA (March 2002) 101
12. Russell, S., Norvig, P.: 6: Adversarial search. In: Artificial Intelligence — A Modern Approach. 2 edn. Prentice Hall (2003)
13. Ruohomaa, S., Hankalahti, A., Kutvonen, L.: Detecting and reacting to changes in reputation flows. In: Trust Management V. Volume 358 of IFIP Advances in Information and Communication Technology., Copenhagen, Denmark (June 2011) 19–34
14. Nurmi, P.: Perseus – a personalized reputation system. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society (2007) 798–804
15. Jøsang, A., Hird, S., Faccer, E.: Simulating the effect of reputation systems on e-markets. In: Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings. Volume LNCS 2692/2003. (May 2003) 179–194
16. Margolin, N.B., Levine, B.N.: Quantifying resistance to the Sybil attack. In: Proceedings of Financial Cryptography and Data Security (FC 2008), Cozumel, Mexico, Springer (January 2008) 1–15
17. Srivatsa, M., Xiong, L., Liu, L.: TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks. In: WWW '05: Proceedings of the 14th International Conference on the World Wide Web, New York, USA, ACM Press (May 2005) 422–431
18. Fullam, K.K., Klos, T.B., Muller, G., Sabater, J., Schlosser, A., Topol, Z., Barber, K.S., Rosenschein, J.S., Vercouter, L., Voss, M.: A specification of the Agent Reputation and Trust (ART) testbed: experimentation and competition for trust in agent societies. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. (2005) 512–518
19. Kerr, R., Cohen, R.: TREET: the Trust and Reputation Experimentation and Evaluation Testbed. Electronic Commerce Research **10** (August 2010) 217–290
20. Teacy, W.L., Huynh, T.D., Dash, R.K., Jennings, N.R., Luck, M., Patel, J.: The ART of IAM: The winning strategy for the 2006 competition. In: Proceedings of the AAMAS Workshop on Trust in Agent Societies, Hawaii, USA (2007)
21. Fehr, E., Fischbacher, U.: The nature of human altruism. Nature **425** (October 2003)
22. Akerlof, G.A.: The market for "lemons": Quality uncertainty and the market mechanism. The Quarterly Journal of Economics **84**(3) (August 1970) 488–500
23. Kaur, P., Ruohomaa, S., Kutvonen, L.: Enabling user involvement in trust decision making for inter-enterprise collaborations. International Journal On Advances In Intelligent Systems **5**(3&4) (December 2012) 533–552