Trust Objects: Enabling advanced reputation services in the Web of Trust

Authored by Moses Ma and Dr. Rutu Mulkar-Mehta, FutureLab

Submitted to the 5th Rebooting the Web of Trust Technical Workshop as a discussion paper, Boston, October 3-5, 2017

ABSTRACT

This paper describes the principles and design considerations for an online framework to manage reputation within a web of trust. The approach uses both transactional and non-transactional data. Non-transactional data includes both trust primitives such as verified claims as well as indeterminate trust assertions. We will also describe the potential use of incentive tokens for incremental optimization of the eco-system, in a manner that is especially suited for decentralized, self-sovereign eco-systems. Finally, we propose to discuss the complexities of online disagreements and how to resolve and adjudicate them in a pareto-optimal manner.

Keywords: reputation, trust, verified claims, collaboration, innovation, framework, blockchain, decentralized, self-sovereign

Background Statement

Reputation is social concept that is an essential component of social and business networks, because it serves as an optimizing influence on such systems. They have been implemented in e-commerce systems such as eBay, Amazon, and many others, and credited as success factors for these systems (Resnick, et al., 2000a). Several research reports have found that seller reputation has significant influences on on-line auction prices, especially for high-valued items (Houser and Wooders, 2000; Dewan and Hsu, 2001) proving that reputation systems are vital for more effective price discovery.

However, while there have been numerous analyses of how reputation may be computed and managed, there has to date been no systematic approach for implementing reputation systems, nor strategies for self-optimizing reputation management, proposed for decentralized networks. As Friedman and Resnick (1998) have pointed out, because pseudonyms create the incentive to misbehave without paying reputational consequences, this should be an important issue in self-sovereign systems.

This paper describes the principles and design considerations for an online framework to manage reputation within a decentralized network, which we call a "web of trust". This framework would include the utilization of distributed ledgers containing information about persistent trust objects and markers, for tracking transaction outcomes, and generating and tracking token incentives to encourage the collaboration process. And it delves into the question of how reputation is related to trust, image, brand, propensity to reciprocate, and other related concepts. This paper proposes an inter-disciplinary, game-theoretic model for computational trust and reputation based on the psychology and micro-economics these concepts and their relationships.

The proposed approach utilizes both transactional and non-transactional trust data. Transactional data includes a record of failed vs successful transactions, such as the history of successful vs unsuccessful transactions at eBay. Non-transactional data includes trust primitives such as verified claims, as well as indeterminate trust assertions. This paper also shows that it is possible use incentive tokens to drive incremental optimization of the eco-system, in a manner that is especially suited for decentralized, self-sovereign eco-systems. Finally, we propose to discuss the complexities of online disagreements and propose key design considerations for systems that could more effectively resolve and adjudicate disputes, in a pareto-optimal manner.

Understanding Decentralized Systems

First, let us review the key design considerations for decentralized reputation, developed by C. Allen[1] and by A.C. de Crespigny et al[2], at the Spring 2017 RWOT design conference. The de Crespigny discussion paper outlines Project Vouch, a system for establishing and verifying self sovereign identities based on attestations and reputation, independent of any central parties, which attempts to satisfy the following design constraints:

- *Context:* what is the reputation value applicable to? What can be understood about an entity by seeing their reputation value(s)?
- *Participation:* how is it defined who can and can't participate, and who can and can't have a reputation value assigned?
- *Consent:* Is consent required by a user to issue claims or a reputation value against the user? Is consent required to reveal claims or a reputation value of a user?
- *Obfuscation:* To meet consent requirements, how is data that calculates a reputation value obfuscated? Can it be derived or is it perfectly information concealing?
- *Value:* How is the reputation value calculated? How are claims contributing to the reputation value normalized?
- *Performance:* How does the system manage the performance and behavior of the users? How does it manage the performance of the network for speed, reliability, and data integrity? How do users have confidence in this?
- *Sustainability:* How does the system stay relevant over time?
- *Claim lifecycle:* How are claims valued over time? Can they be revoked, and under what conditions?
- *Resilience:* How does the system protect against attacks that reduce the integrity of the reputation value?
- *Legal:* What is the legal environment in which the system sits? Are there potential violations of 'natural' law?

This provides an effective starting point for this paper.

---

[1] http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html

[2] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2017/blob/master/topics-and-advance-readings/ProjectVouch_Peer-attestation-and-reputation-based-identity.md

Trust and Reputation Models

According to P. Herrmann et al., in a paper titled *Operational Models for Reputation Servers,* there are four possible operational models for reputation systems, termed the Voting Model, the Opinion Poll Model, the MP Model and the Research Model. The authors propose that it is possible to use two main axes for categorizing reputation servers. The primary axis distinguishes between who performs the evaluation of a subject's reputation based on the available information. The second axis distinguishes how the information is collected by the reputation service prior to collation and publishing the reputations. The reputation system then simply collates these values.

When these two axes are combined together we get the 2x2 matrix shown in the figure below. Each of the four combinations has been given a name for ease of reference and this name is a metaphor to depict the primary characteristics of the operational model:

	Data Push	***Data Pull***
Actor evaluation	Voting model	Opinion Poll model
Reputation evaluation	MP model	Research model

Fig. 1. The four operational models

In the Voting model, the actors evaluate the reputation of subjects, using their own information and experience, compile this heuristically, and then forward their decisions to a central voting server. The role of the voting server is simply to collect messages that arrive, collate and summarize them (again using a simple publicly available algorithm), and then publish the results when asked. E-bay is one example of this type of reputation server in use today. It is quite powerful and accurate, but with full transparency, the system often leads to disputes and requires an efficient dispute resolution methodology.

In the Opinion Poll model, the reputation server actively collects reputation data from the participants. Each actor performs its own evaluation about the reputation of a subject, using his or her own internal assessment, and based on its experiences of performing transactions with the subject. In real life, people do this sort of evaluation all the time – like remembering which shops or partners are good or bad – and our brains are generally optimized for this task. However, these internal assessments are not extremely accurate and are based more on affective memory than keeping an accurate record.

In the Research model, the reputation server actively searches for information about subjects, and then evaluates it and publishes the results. The operations of the reputation server are complex and difficult to engineer. Not only does the reputation server have the problem of finding the actors, as in the Opinion Poll model, but also it has to determine what raw information to solicit from them and how to process and evaluate this in order to compute the reputations of the subjects. Such processes and algorithms are likely to be proprietary and commercially valuable. Standard & Poor and Dun & Bradstreet are examples of such reputation services.

In the MP Model, in which *MP* stands for *Member of Parliament,* a person elected to the UK House of Commons to represent a constituency. MPs should represent their constituency, but often they do not. When it comes to voting on issues in the House of Commons, they either usually follow the party line, or if a free vote is allowed, on such

issues as capital punishment or hunting with dogs, they follow their own conscience. So even if constituents have sent them lots of letters imploring them to vote one way, they may quite freely decide to vote the opposite way. A reputation server following the MP model, will be sent raw data about subjects by the actors. Some of this may be data about transactions an actor has undertaken with a subject, others might be subject reputations evaluated by the actors themselves. The reputation results will primarily be based on the subjectivity of the MP server while the point of view of the actors submitting information may be ignored.

The Opinion Poll model is generally considered the most trustworthy and reliable, since the individual reputation scores have been calculated by a large population of actors. Of course, if many actors conspire together to inflate or deflate a subject's reputation, this is very difficult to protect against. The Voting model should provide the next most trustworthy set of results. The individual reputation scores have similarly been determined by many actors as in the Opinion Poll model, and therefore it should be difficult to skew the results, though it is not impossible. Because the list of actors is not public, it is not possible to independently validate the composite reputation results, nor is it possible for another reputation service to repeat the results.

Principles of Non-transactional Reputation

In general, all of the aforementioned models are based on transactional reputation systems. However, there exists another form of reputation, which is only indirectly transactional, in that they are the assessments of transactional data and hearsay about that data. This form of reputation is much harder to define, but is equally valuable to understand.

We believe there are several important principles that apply to non-transactional reputation systems, which could be used to help govern their design and operation within enterprises and organizations. These are:

- *Reputation is complex.* There is some confusion about the difference between trust and reputation. We believe that trust is a behavior, whereas reputation is a measure of probability of a successful transaction or collaboration. Branding is a form of reputation, in which the consumer is trained to trust a company's brand.
- *Reputation is transitive.* This means that a reputation rating has to be modified or weighted by the party providing the reputation rating. If the provider of a reputation rating itself cannot be trusted, then that rating must be weighted by the probability of that source being inaccurate.
- *Reputation is a convolution.* This means that a reputation probability is not additive or multiplicative, it must be obtained via the convolution of cumulative probability distribution functions provided by both transactional and non-transactional data. Also, the alignment of two or more unrelated reputation systems could provide more confidence in the reputation of that potential partner, if the dispensing reputation systems are in agreement.
- *Reputation is a narrative.* Since reputation varies with time, it is dynamic and always changing. Thus, reputation requires hearing the full story before rendering judgment.
- *Reputation exists in the context of community.* Any given context will have specific factors for what is important in determining reputation.

- *Reputation is a currency.* While you can't change reputation directly, reputation can be used as a resource.
- *Reputation engines should be handled with care.* Doing reputation wrong will lead to organizational squabbles and ineffective incentivization, so proceed with caution.
- *Reputation in the virtual and real world are linked in a complex manner.* It's important for enterprises to allow reputation engines to (i) connect to the real world, (ii) be fully recompilable to slowly conform to and align with the organization's culture and established incentive systems, and (iii) be dynamically modifiable by participants in an organic manner.
- *Reputation is a dynamic social process, not a static formula.* For example, eBay has the most simple model possible, but at the same time, it's the most effective online reputation engine in the world. They did not over-engineer it, but instead, allowed it to be organic and adaptable.
- *Reputation must be kind and flexible, but transactions should be persistent and immutable.* Once the system has recorded a transaction, it cannot be deleted by any of the parties to the transaction. However, reputation systems should be kind to its users and allow for improvement, based on adherence to desired behavior.
- *Reputation accuracy is empowered by managed partial transparency.* Although transparency is a basic requirement for trading systems, social mechanisms require limited transparency to lubricate interaction. For example, when the boss posts an idea and wants to get honest feedback free of "brown-nosing", posting an idea anonymously makes sense. Also, in a social environment, negative feedback may be difficult to provide without some cover of anonymity. This anonymity may need to be removed at some later stage, so the concept of carefully managed partial transparency is a powerful one.
- *Reputation is all about people.* It is important that any reputation server deployed be simple and deep. It must be simple because abstractions and complex nomenclature will be rejected by any mainstream user base, beyond the technically adept R&D team. It must be deep because it has to mimic or surpass the rich interpersonal and interactive cues and rituals used by human beings, that was developed over ten thousands years of civilization.

In general, fully automated reputation engines that are not designed on these principles will eventually fail. An example is the current anti-spam infrastructure, which uses a Bayesian scoring system to accumulate information about likely offenders of spam rules. However, those spammers simply use the system against innocent bystanders, by hijacking email reputation and implementing "joe job" attacks on valid email addresses. Those innocent bystanders will then be unable to receive valid email due to the foibles of the distributed anti-spam systems in place.

Computational Reputation

We believe that a self-optimizing reputation framework needs to use adaptively weighted voting to assess trust and reputation. In our model, reputation is defined to be a convolution of transactional and non-transactional data, with associated weighting

based on the trustability of the rater. For purely transactional data, the weighting would be stronger, and for indeterminate trust assertions, the weighting would be weaker.

Each piece of trust data is a vector with two attributes: (i) the probability of successful transaction, (ii) size/scope of trust area – which can be weighted by the trustability of the data source. We collect all of the available data, in order to gauge the width of the distribution to get a sense of the confidence in that reputation score. Thus, a "trust object" can be described by:

$$T_O = f(\overline{P}, \overline{S})$$

where T_O is a trust datum, P is the estimated probability of successful transaction according to this source, S is the size/scope of trust area, and T is the trustability of the data source.

And so, the reputation of users in the eco-system is a convolution of that trust data, weighted by the trustability of that data source.

$$R_U = g(T_1 \text{ x } PR_{RU1},\ T_2 \text{ x } PR_{RU2} \ldots T_n \text{ x } PR_{RUn})$$

Reputation of person "u" is R_U, where R_U is a function "g" of the trust datum of each data object times the probability prediction of 'u' of that trust object –'PR_{RU1}', for each of the trust objects.

This would provide a multi-dimensional vector that defines the probability of a successful transaction over a range of transaction sizes and scope, along with a confidence factor for the type of trust data offered. Therefore, a counterparty could use that vector, estimate the risk for their particular scope or size of transaction, and include reputation as a way to weigh which vendor or contractor to use.

Furthermore, it should be noted that a reputation or track record should evolve and improve over time. Thus, the system should collect an extensive history of such votes and contributions to capture the time varying effects of such performance, rather than a point estimate, like a credit score.

An adaptive weighting strategy could be used to continuously refine and adjust weights… which are just as important as the underlying transactional data. Also, the system might look for a reciprocity or retaliation pattern detector, in order to filter out such effects as the social reluctance to give negative feedback.

Furthermore, these trust values should be automatically normalized. For example, if a particular voter tends to be a "hard grader", we need normalize to correct this bias, so that the impact such hard graders or scam graders, like on Yelp, can be mitigated.

Finally, we believe that all of these reputation scores could be maintained as discrete distributions (DPDs) so that the confidence in a certain reputation score can be accurately assessed with greater speed at any time.

An Open Framework for Reputation Analysis and Computation

There is a need to create standards for non-transactional reputation, primarily to enable business partners to interoperate more effectively over the Internet. For example,

OpenPrivacy.org has provided the industry with an open Reputation Management System, which eventually became http://www.newsmonster.org/. This system included a bias manager and a reputation calculation engine.

Furthermore, inter-system reputation requests and responses should have an accompanying audit trail of external brokers and services used to calculate the reputation. Confidence measures such as the quantity and quality of data used during a reputation calculation should also be part of a reputation framework response. We believe that maintaining full distributions of data will aid in estimating confidence.

Finally, we should map this system to something like the W3C Verifiable Claims standards proposal. For example, a simple claim is proposed to contain only an identity profile, an entity credential, a "claim" in string format, and a signature.

Field	Description
ID	http://example.gov/credentials/3732
Type	["Credential", "ProofOfAgeCredential"]
Issuer	https://dmv.example.gov
Issued	2010-01-01
Claim	ID: [did:ebfeb1f712ebc6f1c276e12ec21]
	ageOver: 21

Fig. 2. A simple claim

Our proposal is simply to add a field to the basic verified claim system, in the form of a "protocol cookie". Cookies were designed to be a reliable mechanism for websites to remember stateful information or to record the user's browsing activity. Therefore, this field would enable the claim to remember stateful information – such as a URI for the claim offerer's reputation rating, or to record the history of entities that have accessed the claim, or to manage visibility settings for the claim, so that disclosure of the claim could be selectively permissioned. The most valuable function of the field would be to provide a URI to an ontology or classification system for the claim to provide metadata about the size or scope of the claim. These would most likely be stored on some blockchain developed in the future, and adding that extensible field now would enable innovation around with verified claims to be pursued in an accelerated manner.

Self-Optimizing Reputation System Design Philosophy

Beyond the principles discussed in the last section, we have developed a design philosophy for the implementation of reputation engines that could be "self-optimizing". The functions of a self-optimizing reputation engine include:

- uses partial transparency to enable more honest feedback
- voting on ideas is weighted by reputation
- reputation weighting is adaptively and dynamically generated
- auto-normalize voting to weed out the outliers, who may consistently underrate (for psychological reasons), or consistently overrate ideas (eg, "shills").
- the system includes "optimization tokens" that promote more effective and truthful rating and reporting by people

- these tokens can also be earned by artificial intelligence systems that act in a manner similar to miners, but not for providing infrastructural services, but rather, optimization services.
- for example, neural network based fraud detection systems could be developed to look for "ratings extortionists", who threaten negative reviews if not provided a discount

Concluding Remarks

This paper has described a system for managing online reputation within a collaboration environment, and discusses how non-transactional reputation can be mixed with transactional reputation data to create a unified, multi-dimensional reputation assessment. Further, we introduce a set of principles for governing the design and operation of such systems, which provide us with a design philosophy. It is vital to collaborate with others to refine and improve this central philosophy, as this model drives our goals for accomplishment.

As this design philosophy integrates knowledge about people and computer programming, it will be challenging to find the prime solution with a truly motivating value proposition. As a litmus test, we will test the design philosophy against a particularly human use case: dispute resolution. Our goal is to understand both the computer programming and human interaction requirements fully, so will design against the goal of managing the nuances and complexities of online disagreements and how to resolve and adjudicate disputes in a pareto-optimal manner.

Pareto-optimality or pareto-efficiency is a concept from game theory that describes a strategy that cannot be made to perform better against one opposing strategy without performing less well against another – it is a mathematical description of optimizing the collective utility function of all parties in a transaction. We believe it will be possible to implement a pareto-optimal dispute resolution process by using "optimization tokens." It is our goal to demonstrate this system as a pilot in the near future.

In sum, the proposed model addresses the design requirements for an online framework to manage reputation within a decentralized network. It delves into the question of how reputation is related to trust, and other related concepts. Finally, we have offered an inter-disciplinary, game-theoretic model for computational trust and reputation based on the psychology and micro-economics these concepts and their relationships, which should help to accelerate, simplify and optimize online reputation systems.

Project co-investigators:

Moses Ma and Dr. Rutu Mulkar-Mehta, FutureLab
For more info, please email moses.ma@futurelabconsulting.com