Communications of the Association for Information Systems

Volume 37 Article 41

10-2015

The Design of Trust Networks

Levent V. Orman
Cornell University, orman@cornell.edu

Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

The Design of Trust Networks

Levent V. Orman
Cornell University
orman@cornell.edu

Abstract:

One can use trust networks to find trustworthy information, people, products, and services on public networks. Hence, they have the potential to combine the advantages of search, recommendation systems, and social networks. But proper design and correct incentives are critical to the success of such networks. In this paper, I propose a trust network architecture that emphasizes simplicity and robustness. I propose a trust network with constrained trust relationships and design a decentralized search and recommendation process. I create both informational and monetary incentives to encourage joining the network, to investigate and discover other trustworthy agents, and to make commitments to them by trusting them, by insuring them, or even by directly investing in them. I show that making the correct judgments about trustworthiness of others and reporting it truthfully are the optimum strategies since they reward the agents both with information by providing access to more of the network and with monetary payments by paying them for their services as information intermediaries. The extensive income potential from the trust connections creates strong incentives to join the network, to create reliable trust connections, and to report them truthfully.

Keywords: Trust Network, Social Network, Design Architecture, Search, Incentives, Trust Context.

This manuscript underwent peer review. It was received 11/06/2014 and was with the authors for 1 month for 1 revision. Yajiong Xue served as Associate Editor.

1 Trust Networks

Trust has emerged as a critical issue on public networks where interaction with strangers is the norm. Public networks are routinely used to search for information, people, products, and services, yet these resources' trustworthiness and the reliability are difficult to establish, especially when the relationships are ephemeral and transient. Formal contracts are sometimes useful, but enforcing them is costly, making them unambiguous is difficult, and, for small personal transactions, they may be too complex to create and enforce. A trust network, if designed carefully, may provide an alternative and superior solution. Moreover, as critical as it is, trust is only one of several problems with search on public networks, and a trust network built on top of a public network infrastructure may alleviate all of them (Easley&Kleinberg, 2010; Josang, Ismail, & Boyd, 2007). Searching for information in public networks has four major problems:

1. Search on public networks does not include the "dark Web". Databases of people, products, and services can be searched only through documents that serve as front ends to those databases. Searching for a qualified surgeon, for example, returns documents containing various surgeon-related information and not a specific surgeon qualified and trustworthy for a specific goal. Consequently, humans need to manually process the returned documents (Malaga, 2001).
2. Unlike question-answering systems, searches on public networks do not find the relevant parts of a document in response to a question but return the whole document. In searching for a surgeon, for example, the search does not find the description of a specific surgeon in the document that meets the searcher's needs but returns the whole document if it contains any information on surgeons (Richardson, Agrawal, & Domingos, 2003).
3. Search is personalized only to the question and not the questioner. Documents are returned to the extent that they are relevant to the search regardless of the questioner. Occasionally, a questioner's previous searches are used to create a context, but beyond that limited personalization, there is little effort to use the searcher's characteristics to guide the search. In fact, there is considerable evidence that most consumer goods and services are valued subjectively and can benefit from personalization (Adomavicius, Tuzhilin, & Zheng, 2011; Borgatti &Cross, 2002).
4. Finally and most importantly, the trustworthiness and the reliability of the resources located through a search is uncertain. The reliability and relevance of Web documents are often computed from the link structure of the network, but beyond that limited computation, there is little effort to the determine the reliability and the trustworthiness of the documents retrieved, the specific information contained in those documents, or the resources referenced in those documents. A personalized trust computation is even more beyond reach (Andersen, 2008; Moldoveanu & Baum, 2011).

An alternative search strategy involves designing a trust network and combines the advantages of search, recommendation systems, and social networks. In this environment, a network of trust paths connect people to the things they trust, whether they are other people, information, documents, products, or services. Typically, trust is a relationship among humans, but I extend the concept to include not only humans but also information, products, and services. Such an extension leads to an integrated treatment of all by combining the advantages of search that deals with information, recommendation systems that typically involve products and services, and social networks that typically link humans. Trust in other people will create a social network; trust in products and services will constitute a recommendation system; and trust in documents and other information sources will lead to personalized search. I combine all three components through a search strategy that follows trust paths transitively to locate trusted people, information, products, and services (Andersen, 2008; Huang & Fox, 2006).

To succeed in building such a trust network:

1. The trust network must be easy to understand intuitively by humans, and it must be easy to search computationally by machines. Yet, it needs to be precise and effective in locating trustworthy people, information, products, and services (Josang, Ismail, & Boyd, 2007).

The trust network must be beneficial to all participants. Building a trust network, like all social networks, requires the participation of a large number of people. There must be incentives for people to join the network, to reveal their characteristics and preferences, and to investigate the trustworthiness of other

people, information, products, and services, and to make the results of those investigations available to others in terms of ratings and recommendations (Avery, Resnick, & Zeckhauser, 1999; Feldman, Lai, Stoica, & Chuang, 2004).

The trust network has to be correct, robust, and scam resistant. It should provide incentives against lying, cheating (alone or collectively with others), forming false or multiple identities, providing false evaluations and recommendations, or defecting on transactions (Ba, 2001; Dellarocas, 2003).

Satisfying all three criteria simultaneously is a difficult objective to achieve. Most existing networks quantify the degree of trust rather than treat it as a binary variable, which leads to great difficulty in interpreting the value attached to a trust relationship, especially when multiple agents may be interpreting trust values differently. It leads to even more semantic difficulty when such trust ratings have to be aggregated into a single value. It is not at all clear what it means to trust someone at a level of 0.6 out of 1, and even less clear when that value is aggregated from multiple sources rating this agent. Moreover, most existing networks ignore the context of trust and the constraints that might be attached to each trust relationship (Huang & Nicol, 2010; Malaga, 2001). But trust is highly contextual. One may trust her doctor for medical advice but not for financial advice. One may trust a foodie friend for restaurant choices but not for political decisions. That leads to highly imprecise responses from existing trust networks. Ironically, quantification and imprecision are actually closely related. The need to quantify trust is often a result of the fact that context and constraints are ignored, and, consequently, the fact that trust occurs in some contexts and under some conditions but not others has to be characterized as uncertainty. In fact, if enough information is available about context and constraints, one could often identify trust deterministically (Golbeck, 2010; Huang & Nicol, 2010; Richardson, Agrawal, & Domingos, 2003).

The literature features two standard approaches to trust, and both of them have problems with incentives. One approach aggregates all the trust incoming to an agent to compute a universal trust, also known as reputation, as in Google's Page Rank system or eBay's merchant reputation system. This approach often fails to provide incentives to participate in the system by ranking others since ranking is a service to others and does not provide a return to the ranker (Feldman et al, 2004; Malaga, 2001). In this approach, there are also incentives to provide false positive evaluations to friends and allies and fake negative evaluations to enemies and competitors for financial benefit since there is no cost to doing so. There is considerable evidence of fraud. As much as 20 percent of all reviews on TripAdvisor may be fake (Hancock & Gonzales 2015; Malaga, 2001; Moldoveanu & Baum, 2011; Streitfeld, 2011; Wang & Benbasat, 2008). The existing solutions involve mostly honor systems with some investigation of reported cheating and some attempts to block them. Most of these systems are vulnerable to manipulation as in all search, recommendation, and rating systems (Andersen, 2008; Malaga, 2001). They are all monitored manually to eliminate bad evaluations and fraud. This approach does not provide incentives to participate and evaluate others honestly except as a public service (Malaga, 2001; O'Donovan & Smyth, 2005). More importantly, aggregate ratings are not personalized. They consider all reviews equally relevant to all, and, hence, their reviews may only be relevant to a fictitious average consumer (Dellarocas, 2003; Malaga, 2001; Pavlou & Geffen, 2004; Wang & Benbasat, 2008). On the positive side, this approach is resistant to defection because a good reputation is valuable and fragile: since a few bad evaluations can ruin a global reputation, the participants with good reputations do not have an incentive to defect (Malaga, 2001; O'Donovan & Smyth, 2005).

The second approach to trust attempts to find a path from one agent to another transitively, which leads to a personalized trust, such as the Advagato and Trustlet trust networks for open source software developers (Golbeck & Hendler, 2006; Guha, 2004; Massa, 2008). This approach provides incentives to rank others since that is how an agent is linked to others worthy of trust, but then such a personalized trust creates incentives to cheat and defect since the impact of defection is localized (Ba, 2001; Braynov & Sandholm, 2002). These systems are not readily vulnerable to manipulation since each attempt to manipulate effects a small segment of the network. On the negative side, this approach encourages individuals to create multiple identities: each individual builds a reputation in a different niche and each individual has an incentive to defect once they build a good enough reputation. They can benefit from defecting from their commitments because those defections effect their reputation only locally and are not as catastrophic as the destruction of a global reputation (Feldman, Lai, Stoica, & Chuang, 2004; Massa, 2008; Ziegler, 2009).

To remedy these problems and to satisfy all three criteria I set forth, I adopt four main strategies that each draw on tools from a well-established research area; namely, social networks, ontologies, trust, and incentives. Extensive literature in each one of these areas exists, but I combine tools from all of them into

a single design architecture. First, I use trust networks to propagate trust to new agents over a social network. Extensive literature on computing and interpreting connections in social networks that is relevant to trust networks exists (Braynov & Sandholm, 2002; Golbeck & Hendler, 2006; Huang & Nicol, 2010; Kuter & Golbeck, 2010). I start with a binary trust network and later expand it to multiple levels of trust. A binary trust network computes only complete trust or no trust for every agent. It seeks to find a path of consecutive trust relationships from one agent to the other, and, if it does in a limited path-length requirement, it declares trust and terminates; otherwise, it concludes no trust. It compensates for this simplicity by introducing context and constraints to each segment of the path. The study of context and constraints is the second area relevant to this work, and it relies on the literature on ontologies. I use the ontology literature on classification, and inference with context, to incorporate context into trust networks (Huang & Nicol, 2010; McKnight, Choudhury, & Kacmar). Third, an extensive literature on trust in automation that focuses on the formation of trust in existing artifacts exists. I focus on designing an architecture that facilitates and even guarantees trust and not on explaining the trust formation in existing architectures. However, understanding the concept of trust, classification of trust types, and the impact of context on trust are relevant to this discussion (Benbasat, Gefen, & Pavlou; Pavlou & Geffen, 2004; Swaminathan, 2010; Wang & Benbasat, 2008). Fourth, incentives are critical in building trust networks both to encourage participation and also to discourage dishonesty and nonperformance (Ba, 2001; Pavlou & Geffen, 2004; Söllner, Hoffmann, Hoffmann, Wacker, & Leimeister). My approach provides both informational and monetary incentives. The information received from the system is proportional to the quality of one's trust connections, so there is an incentive for individuals to participate in the network, make an effort to establish reliable trust connections, evaluate their trustworthiness correctly, and report it honestly. In addition, I establish a payment system to make it much more profitable to participate in the network, to discover trustworthy partners, to evaluate them, and to report those evaluations honestly since the payments received are directly proportional to the quality of the trust connections one makes. One has the opportunity to generate even more income by investigating and learning about other agents and insuring them against defections or even directly investing in them. Incorporating search, insurance, and investments in the same trust network creates strong incentives to be truthful to maximize one's benefits from all of those opportunities since defecting and cheating closes off all of those opportunities. I discuss these four strategies and their implementation details in Sections 2 and 3.

2 Network Design

A binary trust network is a directed network where an edge from node A to node B implies that A trusts B. It has the advantage of tremendous simplicity: agents either trust each other or not. Searching the network involves a starting point, which corresponds to a searcher, and an end point, which corresponds to a response, and the search is about finding a path that connects the two. Any search will start with the searcher node and will simply follow the trust edges until it finds the required information, people, products, and services. The network assumes that trust is transitive; that is, if A trusts B, and B trusts C, A should trust C. I expand this model later in this section to include multiple levels of uncertain trust. Information, products, and services will always be the end points in such a search since, unlike humans, they can't trust others. Humans, on the other hand, can be either the target of a search or merely intermediaries to create a trust path to the target. The search will always employ a breadth-first strategy and will always find the responses that are closest to the searcher in the network arguably because those are the most trustworthy. The search will terminate as soon as an appropriate response is found, or it will terminate with failure if it cannot find an appropriate response within a pre-specified number of steps. A query may request a single response, the top k responses for an arbitrary k, or even dynamically control the search by checking each response and deciding to continue or terminate at each step. There is evidence from the literature on social networks that only several steps may be sufficient to locate most reasonable paths if they exist, so a reasonable strategy is to terminate searches with failure after a small number of steps. This is a simple strategy to ensure termination since loops are very common in trust networks and loop detection in a large network can be very inefficient. Moreover, there is some evidence that very long transitive trust paths may not be as reliable because trust is not purely binary and there are degrees of trust. But trust can be approximated as binary as long as the paths are short, and low levels of trust are not useful (Golbeck, 2010; Guha, 2004; Kuter & Golbeck, 2010).

Trust relationships are not universal but contextual and constrained. They have context and constraints attached to them, indicating when the trust relationship holds. Searching through the trust network requires testing the context and constraints at each step to ensure that each trust relationship on the path holds for that search. I implement context as a tag attached to each edge, where a tag is an attribute-

value pair separated by an operator, such as context=medicine or context=finance to indicate the domain of trust. Furthermore, other arbitrary tags can be attached to each edge to further constrain the trust relationship. For example context=medicine, specialty=orthopedics will restrict the trust to the orthopedics specialty of medicine. Context=medicine, specialty=surgery, specialty ≠ cardiology will restrict the trust to all surgery except cardiac surgery. Multiple trust links between two nodes is possible capturing multiple contexts or multiple disjunctions of constraints (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005; Conesa, Storey, & Sugumaran, 2008)

A query in this environment is merely a collection of tags, and each edge of the network is also a collection of tags. At a basic level, processing the query involves starting with the searcher node, and following all edges that match the query tags until a response is found. A search for a trusted surgeon, for example, will start with the searcher's node and follow all edges tagged with (context=medicine, specialty=surgery), which take the search to consecutively trusted nodes for surgery until a surgeon node is reached. As simple as it appears, processing such queries in a large and complex network requires an elaborate search strategy and a sophisticated support structure. Six major issues exist in processing such queries that distinguish them from query processing in standard ontologies.

2.1 Issue One

The network needs to distinguish between the intermediate nodes and the target nodes in a search. Target nodes are the responses to the query, and they have to match it exactly. Intermediate nodes guide the search to the target, and they only need to imply (subsume) the query. Consider the query (context=medicine, specialty=cardiac surgery) to search for a trusted cardiac surgeon. The edges with the constraints context=medicine, specialty=cardiac surgery will certainly match the query and lead to target nodes. But an edge with the constraints (context=medicine, specialty=surgery) is not irrelevant to this query. Since surgery subsumes cardiac surgery, although it is not an exact match and will not be returned as a response, the edge will lead to a node that can be trusted with advice on surgery. In other words, if someone is trusted for their opinion and recommendations in all surgery, then they are certainly trusted for their opinion and recommendations in cardiac surgery even though they are not an exact match to be returned as the response. Consequently, we can summarize query processing as following all edges that imply (subsume) the query until an edge is reached that is implied by (subsumed by) the query. This problem of semantic containment requires using an ontology to identify concepts that subsume other concepts. For example, the following snippet of an ontology would be essential to identify the hierarchy of concepts that subsume each other in medicine:

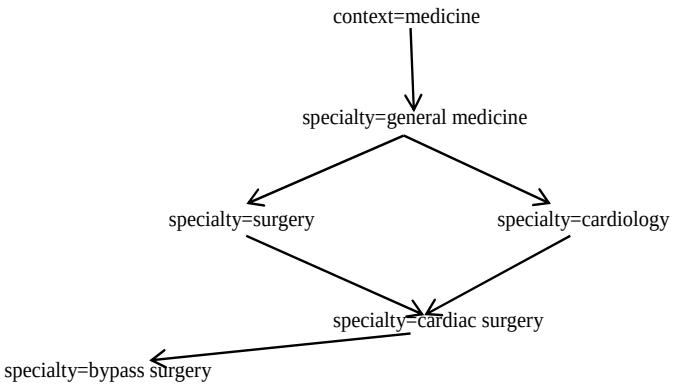

Figure 1. A Snippet of an Ontology Identifying a Hierarchy (Lattice) of Concepts in Medical Specialties

Given such an ontology, any query involving specialty=cardiac surgery will follow all those edges that are constrained to higher concepts such as specialty=surgery since they are implied the query but will return only the nodes that are constrained to lower (or exactly matching) concepts such as specialty=bypass surgery or specialty=cardiac surgery since they imply the query (Fensel, 2001; Middleton., Shadbolt, & Roure, 2004)

Example: given the ontology of Figure 1 and the query (context=medicine, specialty=cardiac surgery), the following network will return only the node D since all paths are implied by (subsume) the query, but only specialty=bypass surgery path implies (is subsumed by) the query (see Figure 2).

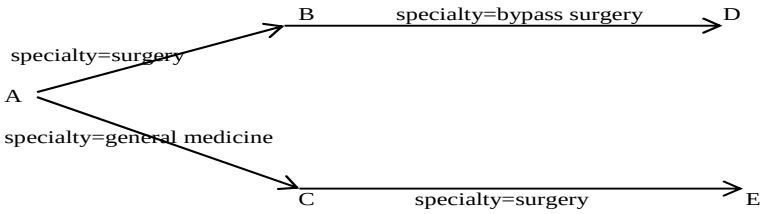

Figure 2. A Sample Network and a Query

2.2 Issue Two

Trust networks do not have to be purely binary. For some applications, a higher level of detail including the degree of trust may be necessary. Trust levels can be included in the network as constraints attached to each edge, such as trustlevel=high. Then the queries can simply add a trust level requirement such as trustlevel>medium, which will exclude all paths with lower trust levels. It is possible to further expand the trust levels to numeric values or even do computations with them. However, most practical applications do not require such detailed computation of trust values but merely some assurance of high trust, which can be done by simply restricting paths to such high level trust values. (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005; Conesa, Storey, & Sugumaran, 2008; Golbeck, 2010; Huang & Fox, 2006; Pavlou & Geffen, 2004; Richardson, Agrawal, & Domingos, 2003).

2.3 Issue Three

The query may contain terms from multiple ontologies involving independent concepts. They do not necessarily lead to empty responses. The basic rule of query processing is for each edge along a path to take the conjunction of the query with the edge constraint and push forward the resulting query to the consecutive edges in the network until the query is satisfied. Missing concepts at each step are assumed to be universe, and the conjunction leaves those terms unchanged (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005)

Example: given the ontology of Figure 1, a standard location ontology, and the query (context=medicine, specialty=cardiac surgery, location=LA), the following network will return only the node D since all edges subsume the query, including the edge (specialty=surgery), with (location=universe) assumed when location term is missing from a constraint. Only the BD edge is subsumed by (implies) the query and, hence, only D is returned as response (see Figure 3).

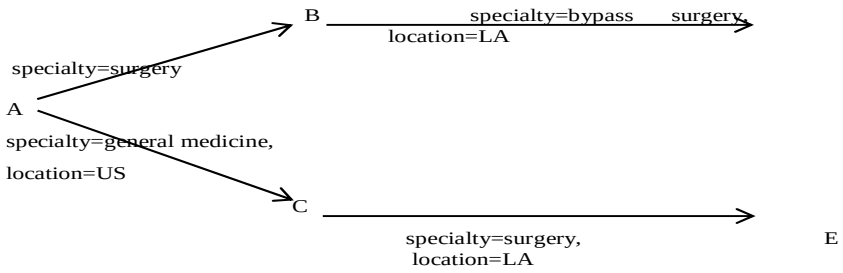

Figure 3. A Sample Network and Query

2.4 Issue Four

Trust networks are heterogeneous and contain nodes corresponding to people, information, products, and services, and we need to relate those heterogeneous nodes to each other for an effective search. For example, a search for a trusted cardiac surgeon cannot only follow the surgeon nodes since many other medical practitioners, researchers, journalists, and even some patients may be able to provide trustworthy information, advice, and recommendations as valuable trust connections. Yet, ontologies tend to be homogeneous relating only the concepts in a context to each other primarily through generalization-specialization hierarchies. To remedy this problem, I expand ontologies to include not only concepts in a context but also people, information, products, and services from other contexts as subtypes of concepts

that define and describe them. A (person=cardiac surgeon), a (service=cardiac surgery), or a (product=stent), for example, will all be declared as subtypes of the concept (specialty=cardiac surgery) for our informational purposes. This extension allows searches for specific people, information, products, or services to identify all relevant nodes that are trusted with the concept that defines those resources and to include them in the search process as intermediate nodes. This extension is a major extension of ontologies to include documents, products, and services as subtypes of concepts that describe them. These nodes relating to physical objects are always leaf nodes because they do not describe or generalize other nodes (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005; Orman, 2015)

Example: Figure 4 extends Figure 1's ontology with people, products, and services:

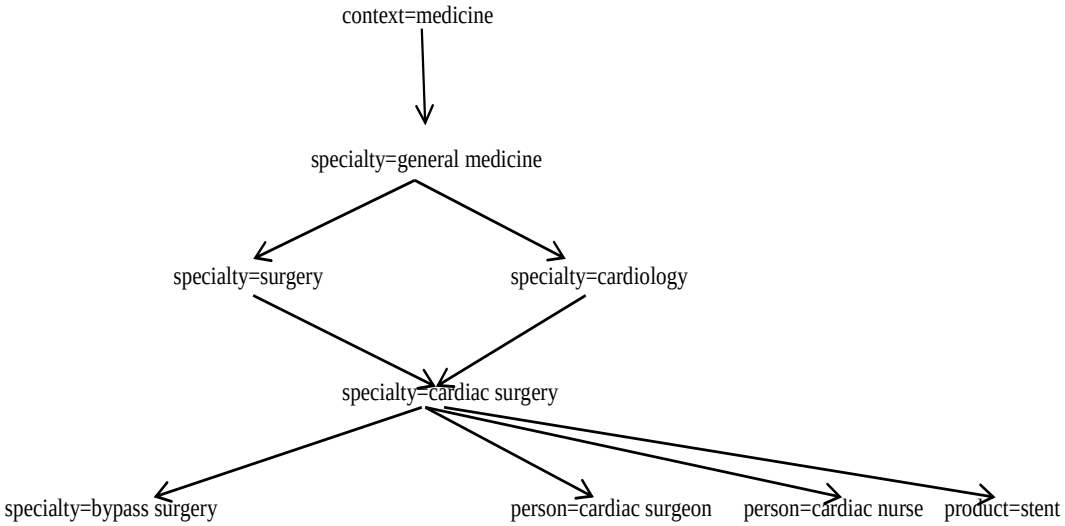

Figure 4. An extended Ontology

Also given the following network of Figure 5, similarly expanded to include persons, products and services, are trusted for their intrinsic physical characteristics not merely as place holders for their opinions and knowledge. Note that A trusts B for his knowledge and judgment on surgery, yet B trusts D as a cardiac surgeon for his physical characteristics as a person in the practice of the profession.

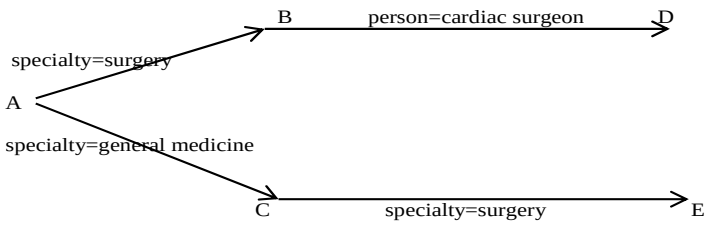

Figure 5. An Extended Sample Network

The query (context=medicine, person=cardiac surgeon) will return node D since, using the extended ontology, all paths are implied by the query, but only the (person=cardiac surgeon) path satisfies the query.

The nodes corresponding to physical objects will be restricted to leaf nodes in the trust network also as in ontologies since, as physical objects, they cannot trust other nodes. If there is a need to treat a person both as a provider of a service and as a provider of information, two separate links are needed. For example, a cardiac surgeon D who not only practices cardiac surgery but also is a source of trusted information to B will lead to two links from B to D: (person=cardiac surgeon) as above but also the link (specialty=cardiac surgery). This will allow the node to be trusted for not only the practice of surgery but also for advice and recommendations, and it will also be able to further trust other products and services since it is at higher level of abstraction in the ontology.

Moreover, one can further limit the trust links by the criteria used to evaluate them. For example, one may trust a cardiac surgeon if the primary concern is this individual's bedside manners or the surgical outcome by using the criteria tag as criteria=bed-side-manners or criteria=surgical-outcome, respectively. Such a link would not be a match if a query listed different criteria such as the waiting time for surgery or explicitly excluded bed-side-manners. If no criteria is listed in the query, all such links would be included because of the default universe assumption. One can also exclude certain criteria from trust links if, for example, the cost is explicitly irrelevant to trusting a cardiac surgeon as in criteria≠cost. Such a link would not be a match if a query explicitly included the criterion of cost as a trust factor.

2.5 Issue Five

The network can process queries in parallel quite naturally over many nodes to gain considerable efficiency in response time. The parallelization is straightforward since each node faces a different task independent of other nodes in processing the query. Each node receives a query and pushes it to its trusted partners if the query is subsumed by the constraint on the link to that partner. The query is modified before pushing it down by taking the conjunction of the query with the constraint on the link. If the query is satisfied by the link, an answer has been found and the node is returned. At each step on the path, the nodes do not have to wait for each other but can do their processing in parallel. Consequently, the response time is independent of the number of nodes in the network: it depends only on the length of a path and the number of constraints that need to be tested at each step by each node. If we limit all search paths to a length of s, that also limits the response time to s evaluations of constraints. If a node has m trusted partners maximum, at most mi constraints need to be tested at each step I, but m^{i-1} parallel tests can be conducted by m^{i-1} nodes, which will lead to at most m sequential tests at each step. Then the time complexity is bounded by sm [30].

Example: given the same ontologies and the same trust network of Figures 4 and 5, the query person=cardiac surgeon will be processed in two steps. In the first step, the node A will take a conjunction of the query with the constraints of each of its trusted partners B and C. Both conjunctions will return person=cardiac surgeon since both constraints subsume the query in the ontology. In the second step, B and C will process the query in parallel for their trusted partners. Both conjunctions will still return person=cardiac surgeon since the both constraints still subsume the query. But the constraint person=cardiac surgeon is an exact match, so the node D will be returned as the answer (see Figure 6).

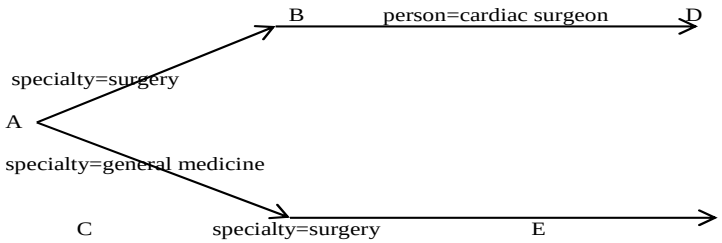

Figure 6. A Query on an Extended Network

2.6 Issue Six

One can greatly improve the efficiency of query processing by detecting and eliminating loops. Social networks are notoriously well connected locally, and any traversing of it will generate many loops that do not contribute to the response. Detecting loops can also be done quite simply and in a distributed manner. Every node can push forward, along with the query, the path that has been followed to get to that node. The receiving node simply checks if any of its trusted nodes are on that path and will lead to a loop; and it will simply ignore those nodes that will. The processing is again limited to s checks since we have limited all paths to a length of s. At each step, m trusted partners have to be compared against at most s nodes that are already on the path. That leads to a time complexity bounded by s2m where m is the maximum number of trusted partners per node.

3 Network Update

Once an agent joins the network and creates its trust connections to other agents, it can use those connections to locate trustworthy information, people, products, and services simply by following the paths

created by consecutive trust relationships. The quality of the information received from the system is directly proportional to the quality of the trust connections one makes. Consequently, there are strong incentives to join the network and to make the best possible trust connections by investigating and collecting information about other agents and by being truthful about whom one trusts. Conversely, it is important for the system to allow dynamic and accurate modification of the trust network as the agents collect more information and execute transactions with other agents. Since agents interact exclusively with the agents they trust, successful transactions and positive information does not impact the network. However, when a failure occurs, the network has to accommodate an accurate and prompt update to the trust relationships. Since a failed transaction may involve many agents, identifying and assigning responsibility and modifying trust relationships correctly can be a challenge. Consider a failed transaction between the nodes A and C as a result of a connection made through node B (see Figure 7).

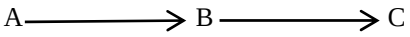

Figure 7. A Sample Network

A now knows that C is not trustworthy and would like to update the network, but C was not one of his trustees in the first place. A can only delete A's own trust of B, but that is not necessarily the right update to the network. It is possible that B is not trustworthy either for its bad judgment for trusting C or because it is in a conspiracy with untrustworthy C; it is also possible that B is generally trustworthy but was simply misled by C. Eliminating its trust of B incorrectly or hastily is also costly for A since B may have many other trustworthy connections that A would like to continue to access. There is a need for a communication system between A and B to judge who is responsible for the failure of C and to what extent (Guha, 2004).

To solve this problem, I introduce a distrust link, which is a temporary link that is used merely to update the network. It is deleted after the network is updated and a new consistent state is reached. In the above scenario, after a failed transaction with C, A will introduce a distrust link to C (see Figure 8).

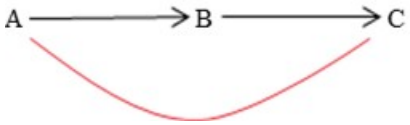

Figure 8. A Sample Network with a Failed Transaction

A would like to eliminate the trust path to C, but it is not clear where the trust was broken—by B or by C. The system will convert this distrust link to a collection of constraints on existing links and notify the relevant agents about those changes. To accomplish this, I introduce a new tag called "path condition" that asserts the nonexistence of a trust edge. Path condition=B / C means B does not trust C, and, when the condition is attached to an edge AB, it means that A will trust B only if B does not trust C. Then the system will convert a distrust edge to a series of path condition, and the relevant nodes will be notified. The nodes will be expected to take action, and the system will modify the network accordingly until consistency is restored. In this particular case, the distrust link will be converted to the following conditional (see Figure 9).

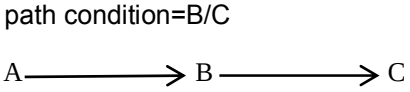

Figure 9. A Network with a Path Condition

B will be notified that trust from A is conditional on its breaking its trust on C. If B does not break its trust on C in a specified time period, the trust edge from A to B will be eliminated and the system will be back in

a consistent state with no conditionals. With longer distrust paths, the procedure involves multiple steps and starts with the end point and moves backwards. At each step, a node will face losing trust unless it breaks its trust to the consecutive node. Consecutively moving backwards in a distrust path identifies possible conspiracies and assigns responsibility for a transaction failure. If B and C are in a conspiracy in the above example, B will not break its trust for C and B itself will lose the trust of A. If B is not in a conspiracy with C, then it will break its trust link to C to keep the trust of A despite the fact that it may know nothing about the transaction that failed. It does not need to investigate; it only needs to know that, because of a transaction failure, it has to choose between A and C. Distrust links make the network very dynamic and easily updated as trust relationships change but without extensive requirement for agents to investigate trust failures and to try to identify responsible parties for every failure. Such instant updates of trust are critical to preventing multiple defections before detection. More importantly, node C cannot defeat the distrust link from A to C because that distrust link forces B to choose between A and C. If B continues to believe that C is trustworthy despite negative comments from A, then B will lose the trust of A, at least for some types of transactions.

4 Payments and Incentives

For most agents, the dependence of the information quality on the quality of one's trust connections and the threat of losing those connections through distrust links are sufficient incentives to join the network and be truthful about their trust. But not for all. It is perfectly feasible in this environment for some to join the network with no intention of using it to search for information but merely to enter false trust connections either to promote their own products and services or to support their allies and co-conspirators. Such actions are not likely to be very effective since they need to earn trust with some agent before their trust connections can be used and only by those who trust them. Nevertheless, some incentive exists to attempt it. Moreover, one can establish multiple identities and use some of them honestly to access quality information yet abuse other identities to enter false trust data for financial advantage. Again, the payoff from such an attempt would be limited since one needs to establish some trust with at least one non-conspirator before one can use the network to any financial advantage. To eliminate even the limited incentives to cheat, I introduce a payment system. The system ensures that the payoff from defection is always less than the expected benefits from staying in the system at all times and for all nodes. I accomplish this by establishing a commission payment system where the expected commissions by each node grow faster with increasing trust than the possible payoff from a defection.

The network is an information system. It is used not only to find products and services but also to facilitate commercial transactions. We can view it as a personalized advertising system designed not only to locate products and services that are appropriate for each individual but also to execute transactions and process the flow of payments from buyers to sellers. As such, it is valuable to vendors of those products and services, and they can be charged a commission on every transaction executed through the system. Those funds can be used to increase the incentives for all participants in the network to be truthful even if they are not seeking information from the system and even if they had joined the system for the express purpose of corrupting it and even defrauding it. The system relies on the premise that the expected income one can receive from the network by providing truthful information and, thereby, facilitating transactions as an information intermediary always exceeds the expected income that can be generated by lying and defecting on transactions. This premise has to be true both for newcomers and the established members, and it has to remain true throughout the lifetime of a member to eliminate the possibility of defecting on a large transaction after building a reputation. It has to remain true even when participants can create multiple identities and defect on some identities while continuing to build a solid reputation on others.

Each agent will receive a commission from the system every time another agent uses its trust connections in a search that ends in a commercial transaction. The system collects a commission from each transaction completed over the network and uses those funds to pay all agents used in searching and executing that transaction since every transaction uses the trust connections of others to locate the right partners, products, and services and execute the transaction. In effect, by aiding other agents' searches, each agent acts as an information intermediary like an advertising agency and gets paid for its services. The amount of income generated from this activity is directly proportional to the quality and the quantity of one's trust connections—both the agents that trust it and the agents it is trusted by since the agents that trust it channel requests to it from upstream nodes and the agents that it trusts provide responses to requests by channeling requests downstream.

To provide proper incentives, the expected commissions an agent collects by supporting others' transactions should exceed the expected returns from defecting on a transaction. To accomplish this goal, the commission rate should be chosen carefully. Let t be the average transaction amount and c be the commission collected by the system on each transaction as a percentage of t, which leads to a payment of ct per transaction. Let there be m incoming links per node on average in the trust network, m outgoing links, and k transactions per agent per period. Let us consider a single trust link from a node A to a node B. We would like to make sure that the value of this link to B is always more than the value of defecting on a transaction, so defecting and losing the trust link is never the optimum strategy. Out of k transactions initiated by A, k/m of them go to B on average since there are m outgoing links from A. Similarly, for the transactions initiated two steps upstream by the m trusters of A, each one channels k/m of its transactions to A, with a total of k transactions arriving at A, and A subsequently channeling k/m of them to B. Repeating this for up to s steps upstream since we assumed paths of length at most s, we get k/m transactions arriving from A to B for each of the s steps away upstream. For any s step path, the transaction may find its target at any step i ≤ s and terminate the path with a probability pi=1-mi/n since, at each step, each node connects to m nodes out of the whole network of n. Then the probability of a transaction starting j nodes upstream and surviving j nodes to arrive at its destination would be $\prod_{i=1}^{j}(1-p_i)$. The expected length of a path for a transaction arriving at its destination can then be computed by using these probabilities as weights on possible path lengths, leading to $\sum_{j=0}^{s}\prod_{i=1}^{j}(1-p_i)$. Call this u. The expected number of transactions arriving from a node A to a node B would then be uk/m since each step on the path contributes k/m transactions. The expected path length u is difficult to compute in general, but, fortunately, we do not need to compute it for our purposes as I show next.

For each transaction, the network collects a commission of ct and divides it equally among the nodes on the path. Since the expected length of a path is u, from the above analysis, the expected commission for B for any transaction that comes to it through A is ct/u. But the expected number of transactions arriving at A is uk/m from the above analysis, which leads to total commissions of ukct/um = kct/m per period. Consequently, the loss of a trust link will lead to a loss of kct/m per period indefinitely. Taking the present value of that indefinite income stream leads to $\sum_{i=0}^{\infty}(1+r)^{-i}\left(\frac{kct}{m}\right)=\frac{kct}{mr}$ where r is the discount rate. We want to ensure that this value is always greater than t since it is the expected benefit from defecting on a transaction valued at t.

$\frac{kct}{mr}$ >t implies that $\frac{kc}{mr}$ >1 or equivalently c> $\frac{mr}{k}$. $\frac{mr}{k}$ then is the minimum commission to be charged each transaction to ensure that defection is never the optimum strategy for a typical agent.

Example: typical values of m=10 trust connections per agent, r=0.05 discount rate per year, and k=50 transactions per year per agent will lead to c=0.01. One percent commission on transactions is considerably less than what advertising agencies currently charge for completed transactions through online ad clicks, and the trust networks playing the same role with a great deal more reliability and personalization should have a significant advantage over the current advertising models.

This analysis also applies to efforts to inflate one's trust to increase the number of transactions flowing through a node and increase income. Any inflation of trust by manipulating constraints is likely to lead to an increase in defections by the same percentage since the upstream nodes can adjust their trust accordingly by changing the conditions of their trust. Upstream nodes do not have to distrust a node completely, but they can distrust under certain conditions. So, if a node B inflates its trust of C by relaxing a constraint X, then the upstream nodes such as A are likely to reduce their trust by deflating their trust by tightening the same constraint X. Any effort to increase the transactions downstream will lead to an equal percentage decrease in transactions coming from upstream, and the above analysis for all or nothing defection applies equally to efforts to partially mislead.

However, the commission is only the sufficient incentive for the "average" agent. Agents that are not well connected can still have an incentive to defect since defection does not greatly impact their future earnings. Modifying the analysis for an arbitrary agent requires considering an agent that may have only n trust connections rather than the expected m connections. This agent will be more or less likely to receive transactions than its neighbors that are competing for the same transactions. Instead of each getting k/m transactions, the agent with q connections will get kq/m^2 transactions since it connects to only q other agents, but its m-1 neighbors each connect to m agents on average, or approximately m^2. Repeating the same analysis as above, we now get c>m^2r/kq for all q for the commission to be an effective deterrent against defection. Clearly, for the typical agent where q=m, we get the same results as above. But for the

less-connected agents, the commission has to be larger to discourage defection as expected since they have less of a commitment to the system and smaller expected benefits in the future.

Example: for a poorly connected agent with n=5, the commission has to be raised to two percent, and, for a new agent with only n=2 connections, a commission of five percent would be necessary to discourage defection.

To deal with those extreme cases, one can easily devise a variable commission system by charging more risky isolated agents higher commissions and guarantee compliance by all using the sufficient commission computation above. An alternative to a complex variable commission system and charging some agents large commissions is to create an insurance scheme, which I discuss next.

For most agents, the information quality and transaction commissions are sufficient incentives to join the network and be truthful about their trust. But this paper introduces a third set of incentives to further discourage cheating. A more elaborate system of payments will further incentivize participation and truthfulness in the network. Agents will have the option to make insurance payments to the system for some of their trusted agents to demonstrate their extra level of trust on those agents. These payments can be viewed as insurance against the failure or defection of those agents because they are used to guarantee the transactions that those agents provide. If an agent fails to fulfill its commitments, then the funds accumulated for its insurance are used to reimburse the injured parties. The amount of total funds guaranteeing a particular agent's commitments is publicly available information for all those considering a transaction with that agent. Obviously, larger amounts indicate more trust in that particular agent and encourage more and larger transactions. Agents who make insurance payments to demonstrate their trust on another agent receive a commission for every transaction executed by those trusted agents in proportion to their trust payment and the transaction amount. There are two kinds of incentives in this scheme. First, the existence of insurance will increase confidence and will increase the total number and amount of transactions executed, which will provide incentives for vendors to stay in the system and fulfill their commitments because future income expectations are higher. Second, and more importantly, providing insurance for one's trusted partners is a source of income for all agents. By using one's knowledge of other agents and their trustworthiness, one can provide insurance coverage for them like an insurance company and collect commissions in return beyond the regular transaction commissions. Such extra income from the system further incentivizes identifying trustworthy agents, making quality connections, and, as one's trust increases, providing insurance coverage to them. Note that insurance does not encourage risky behavior because it is not paid to the vendor but to the damaged parties because of an unfulfilled promise by the vendor.

In this scheme, agents have the option of making payments into the system, which provides guarantees to the operations of any other agent or a commercial provider of a product or service. The payment amount by any agent towards any other agent reflects the amount of trust by the first on the second, and the total amount of insurance payments provided to support a particular agent is publicly available information to bolster overall trust in that agent. The money is not paid to the insured agent but held by the system, and it is used to pay damages to any party that is injured by a failure of the insured agent. In return, insurers get a commission from every transaction conducted by the insured agent. There are two approaches to implementing this insurance mechanism, distinguished by the contractual basis of this arrangement and the enforcement mechanism. The simplest implementation employs a system administrator who collects insurance payments, investigates claims, and makes payments to injured parties just like any other insurance company. This is a simple implementation, but it violates the peer-to-peer and decentralized nature of the trust network.

A more sophisticated implementation is completely decentralized and employs no central administrator; instead, it relies on many insuring agents investigating any claim against their trusted partners on their own. They make payments to injured parties if they choose to do so. They have an incentive to make payments only because the injured parties are also their trust partners and because they want to keep the trust and confidence of their partners. In this implementation, agents still promise insurance payments to guarantee the reliability of other agents they trust, but there is no central administrator and no actual payments to the network; instead, there is only a promise registered with the network. If a failure occurs, the injured party notifies and asks all insurers of the failing agent for payment. They all investigate independently and choose to pay their share of the damages or not. Those who refuse to pay receive a distrust link by the injured party, as in the previous section, and lose some of their trusting partners and commission income because of this failure. There are three critical issues: first, what is the relevant amount of insurance coverage for each transaction that an agent can count on? Second, what is the

appropriate commission rate on insurance to incentivize agents to actually provide insurance? Third, what is the appropriate commission rate on transactions and insurance to guarantee that insurers will actually honor the insurance payment requests for failed transactions? I consider them in sequence.

First, the total amount of insurance available to an agent is not the relevant amount for a particular transaction. The relevant amount is the total insurance provided by only those one trusts because those are the only ones with whom one can break its trust and effectively punish them if they fail to uphold their promise of insurance. Any other insurer would have no incentive to pay since the damaged party would have no leverage over them. So, for every potential transaction, the system will show the agent initiating the transaction the total amount of insurance provided by only the agents that the initiating agent trusts, directly or indirectly. In case of the transaction's failure and a failure by any of the insurers to pay damages, the initiating agent can enter a distrust link to punish the failed party. Insurance in general is a difficult concept, but it can be explained to consumers very simply in this context. The consumers are told for each transaction how much insurance is available for that transaction provided by the people they trust. That amount is what they can claim for reimbursement if the transaction fails, and their leverage is to revoke the trust link they have with those people who provide insurance if they fail to honor a request for reimbursement.

Example: consider the following trust network where B, C, and D all trust E and each provides $100 insurance to E (see Figure 10).

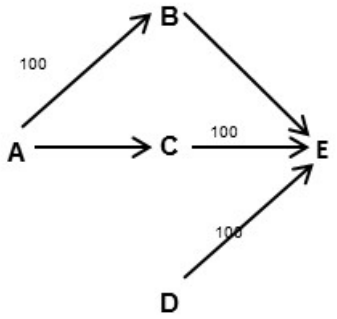

Figure 10. A Sample Trust Network

A gets into a transaction with E worth $100. First of all, the system informs A that there is $200 worth of insurance to cover A's transaction. The insurance D provides is not included in this calculation because A does not trust D and has no leverage on D to compel payment. Let's assume that the transaction fails and A files a report with B and C demanding compensation for its damage of $100 split equally between the two, which results in a demand for $50 payment from each. B honors the request, and probably will eliminate its trust in and the insurance of E, which effectively punishes it by reducing E's future income from potential transactions flowing through B. Let's assume C does not honor the request for damages. Now A has the option of eliminating its trust link to C and effectively punishing it by eliminating all potential transactions flowing through A to C and reducing C's future income. With a large number of agents providing insurance in general, the risk will be widely distributed, and the insurance burden on each agent is likely to be minimal.

Second, what is the appropriate commission on insurance to incentivize agents to actually provide insurance? The answer requires some knowledge of the operations of the insured and an assessment of probability of failure. Insurance is different from other transactions because, first, all insurers— not just the agents involved in facilitating a transaction—get a share of the commissions paid into the system, and, second, all insurers contribute to paying damages in case of a failure in proportion to their insurance commitment. Consequently, insurance commissions involve a larger number of transactions split over many more people, and the damage payments also involve many more people than ordinary transactions. Consequently, risk is much smaller than ordinary transactions. This makes insurance a useful business for insurers but also for customers since it practically eliminates the possibility of a catastrophic loss and, further, for vendors since it increases confidence and the number of transactions greatly. Yet, it accomplishes all of this at a small cost of insurance commissions paid on all transactions. The adequate commission can be computed using a similar analysis as in the previous case where p is the probability of failure and t is the transaction amount leading to an expected loss of pt. That loss has to be covered

collectively by w insurers, which leads to an expected loss of pt/w per insurer. The expected loss has to be less than the expected commission income from that particular agent. Also, if c is the commission rate and t is the amount of transaction, then, for k transactions, kct is the total amount of commission collected. That has to be divided among w insurers, which leads to kct/w>pt/w or c>p/k. This commission rate is a much smaller commission rate than the one for transactions, which demonstrates that insurance is a much more effective way of ensuring compliance than payments on transactions.

Example: for typical values of p=0.01 and k=50 transactions per period, the required commission c is 0.05 percent an order of magnitude less than the one percent needed with transaction payments. With a commission rate of one percent for insurance payments, all transactions, except the most egregiously risky ones by the most isolated vendors, can be guaranteed.

Third, what is the appropriate commission rate on transactions to guarantee that insurers will actually honor the insurance payment requests for failed transactions? This analysis is identical to the analysis of incentives against defection in transactions. In both cases, any defection or refusal to honor promised insurance coverage leads to distrust links and reduction in future transactions and commission payments. The only difference here is the existence of two separate income streams: one from transaction commissions and the other from insurance commissions. Consequently, the expected future income is approximately double the transaction commission income alone as it was in the previous analysis. So, the minimum required commission to prevent defection is still $c > \frac{mr}{k}$, but c here is the sum of two commission streams. So, for typical values, it would still be about one percent. The total of two commissions I suggest above, each typically at one percent, would guarantee compliance by all insurers except the few most isolated ones. Of course, in a distributed insurance scheme, the defection of a few should not significantly impact the damage payments that can be collected by the injured parties. More importantly, a more complex variable commission system can be employed to charge each agent a different commission depending on their connectedness by using the sufficient commission computation above to guarantee complete compliance by all.

There is fourth set of incentives to discourage lying and defections. Further incentives are possible by making direct investments in other agents to support their operations and to take equity ownership in their operations, and get a share of all of that agent's life time income from its network transactions. This concept is similar to insurance in that it allows an agent to support another agent and get a return on its support. But it is different from insurance because payments are made directly to the agent rather than promises made to the system to make payments in case of failure, and the payments are not retractable once made. Moreover, as others make investments into the same agents, the value of a particular investment may be diluted. The incentives to invest are determined by the expectations about the future income stream of that agent and the total investments made into that agent since commissions on all transactions are distributed among the investors in proportion to their investment. The important contribution here is the fact that investments into an agent reduces the incentives for that agent to defect on any transaction because the benefits from any defection will be distributed among many investors as any other income. The costs of defection, which is the loss of future income, is also distributed among many investors, so the net total effect would be neutral, yet the incentives to defect on any single transaction would be much smaller because the transaction itself, whose impact is distributed among many, is much smaller per agent. Consequently, any individual defection is a much smaller event, which reduces one's incentive to defect.

5 Model Validation

I tested and validated the analytical model I develop in Section 4 on an existing trust network—Advagato. The trust network Advagato is a non-profit trust network that contains real data about some 20,000 open source software developers and their trust and judgment about each other's competence and reliability [24]. It is a trust network in the context of software development and its sub-contexts of various programming languages and skills. I downloaded a 10,000 node segment of this trust network into Mathematica software and ran a number of simulations to validate my analytical findings. I split the network into 100 segments of 100 nodes each and sorted them with respect to the average number of connection per node, which led to the first control variable: number of trust connections per node measuring the connectivity of the network. I designed the first simulation was to test how fast a real-world network could reach from one arbitrary node to another by following the trust connections. Figure 11 summarizes the results, which validate the analytical findings that the trust networks can navigate from one node to another quite efficiently—in less than three steps for 92 percent of the nodes in a minimally

connected network of five connections per node. Only the nodes that are extremely isolated with less than two trust connections may typically take longer to reach. I designed the second to determine what portion of the network was reachable in a real-world trust network. Figure 12 summarizes the results, which validate the analytical findings that the nodes are reachable over 96 percent of the time in less than six steps in a minimally connected network of five connections per node. The only exceptions are the most isolated nodes with less than two trust connections.

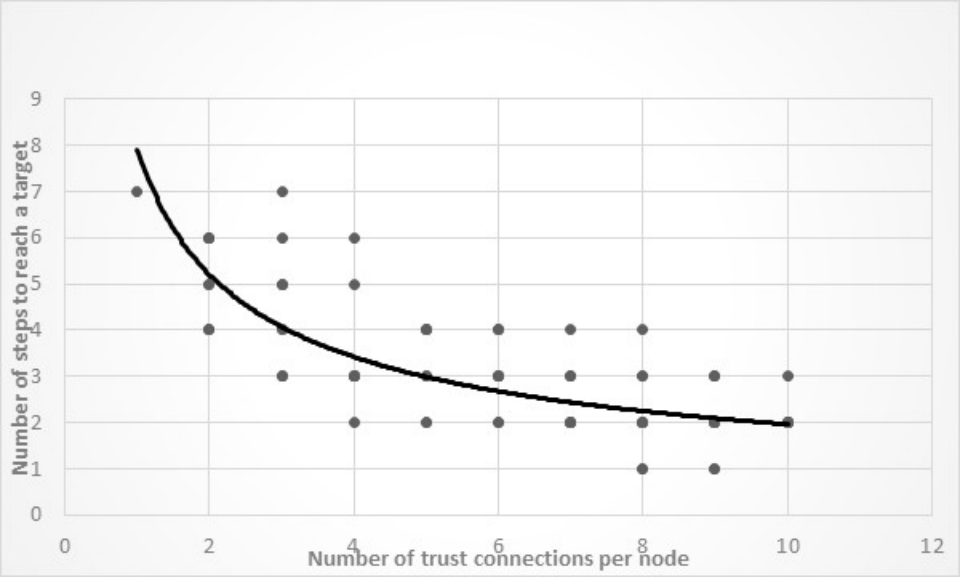

Figure 11. The Number of Steps to Reach an Arbitrary Target as a Function of Number of Trust Connections Per Node

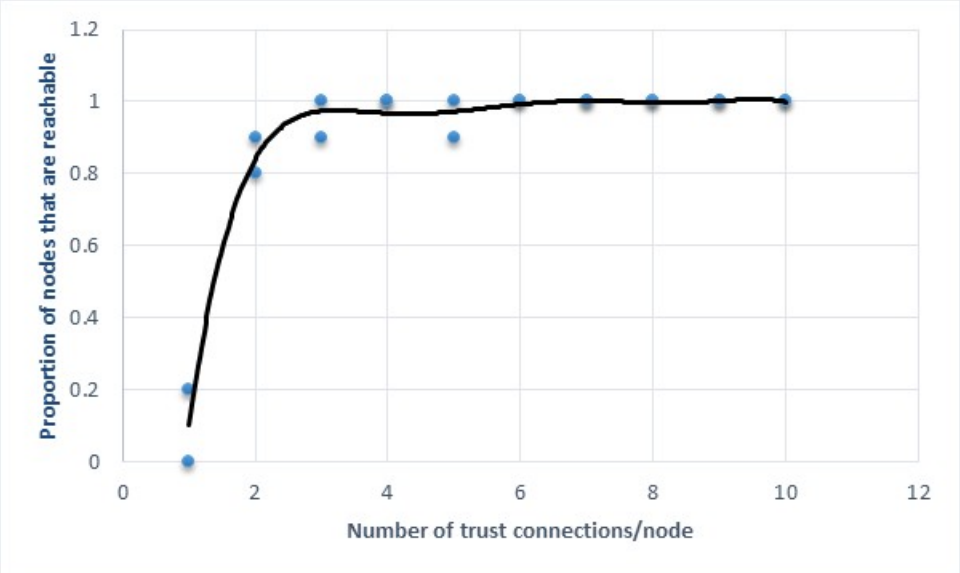

Figure 12. The Proportion of All Nodes that are Reachable from an Arbitrary Starting Node as a Function of Number of Connections Per Node

I designed two more simulations to test the analytical findings about incentives. I generated and ran random sets of 100 commercial transactions on each of 100 samples, which led to a total of 10,000 transactions. Each transaction involved a buyer, a seller, and a transaction value of software services. There were two control variables. The first was the number of transactions per node, which measured how active a node was both in initiating transactions and also facilitating others' transactions. I varied the number of transactions per node from one per year to 25 per year for testing purposes. The second was

the commissions charged per transaction. A certain percentage of the transaction value was paid in fees to those nodes that acted as intermediaries to facilitate a transaction. I varied the commission rate from one to three percent of the value of the transaction, which is similar to the commission rates charged by credit card companies. I used an infinite horizon and an annual discount rate of five percent to computer the total expected commissions. Figures 13 and 14 summarize the results. Figure 13 shows that present value of commissions earned per node as a multiple of the transaction value when the transaction values are normalized to 1. It is shown as a function of activity level measured by number of transactions per node. It validates the analytical findings that the expected commission earnings are typically much higher than the value of an individual transaction, which is the value a node can extract by defecting on a transaction and refusing to pay. The findings justify the claim that an overwhelming majority of nodes have no incentive to defect or cheat and risk future commission earnings. Only a very small percentage of the least-active nodes with less than five transactions per year had an incentive to defect at the level of two percent commission rate. For those, I propose an insurance scheme in Section 4. Figure 14 shows the same findings in a percentage format for easy viewing. It shows that the percentage of nodes with an incentive to defect as a function of their activity level and the commission rate. It shows that only the least active nodes with less than five transactions per year had an incentive to defect and only when the commission rate was less than two percent. For those few nodes, I propose an insurance scheme to make sure that no nodes are excluded from safe transactions.

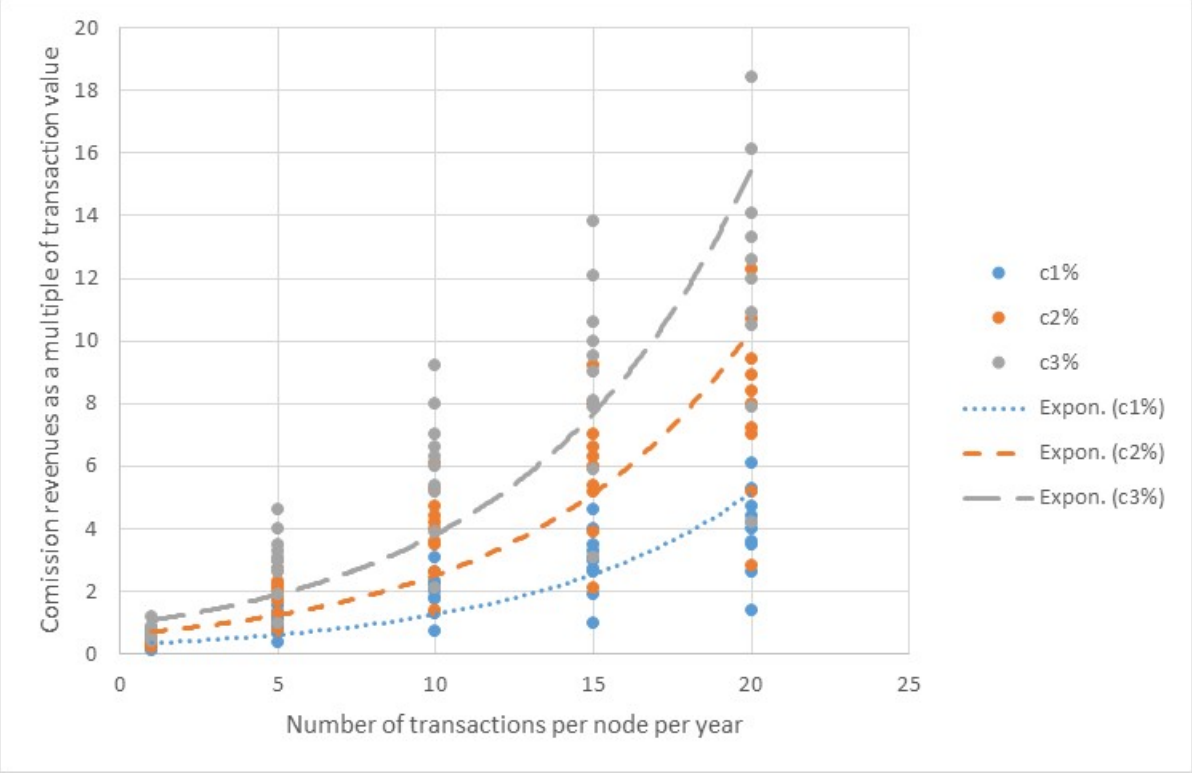

Figure 13. Commission Revenues Per Node as a Proportion of Total Transaction Value for Commission Rates 1%, 2%, and 3% of the Transaction Value. Black Line Corresponds to Value 1 where the Commission Revenues Equals Transaction Value (i.e., The Point at which Defection Becomes Undesirable)

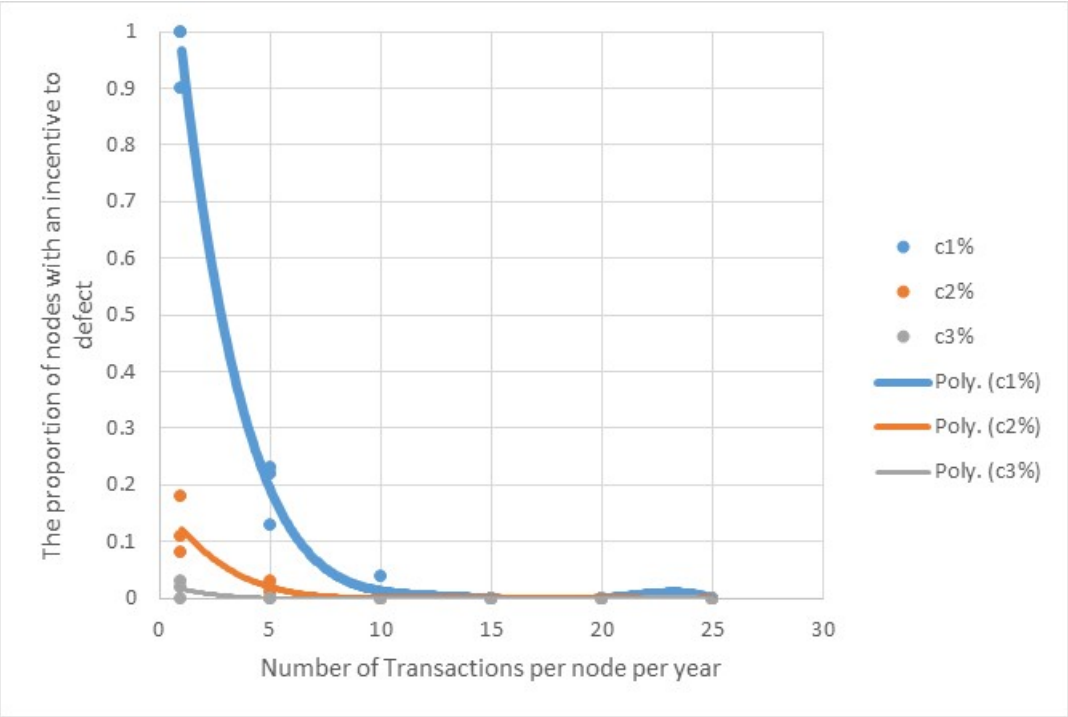

Figure 14. The Proportion of Nodes with an Incentive to Defect as a Function of Node Activity, for Commissions of 1%, 2%, and 3% of the Transaction Value

6 Applications

Trust networks are useful in a variety of applications beyond searching for trustworthy information, people, products, and services.

6.1 Application 1

Money can be radically reinvented and decentralized through an effective trust network. Money is an information system created to value, record, and track economic transactions. It is an information system with minimal semantics and centralized control. Consequently, the monetary system fails to support many transactions directly but requires intermediaries such as banks, brokers, insurance companies, credit card companies, and investment firms, which increases transaction costs greatly (McKnight, Choudhury, Kacmar, 2002; O'Donovan & Smyth, 2005). An effective trust network decentralizes the management of money and enriches its semantics. It can accomplish this by replacing national currencies with private currencies and by creating an automated peer-to-peer network of currency-exchange systems to support complex transactions. Each agent can print its own currency that can be viewed as personal IOUs. They can be used to pay for goods and services to the extent that other agents trust the issuer and will accept its currency as payment. Money is all about trust in the credit of an issuer, and an effective trust system can replace the government as the sole trusted party with many private trusted parties whether they are businesses, non-profit organizations, or even individuals. The network can manage the payments to agents who are not direct trusters of an issuer by finding a trust path where every agent accepts the currency of the previous on the path and a sequence of currency exchanges leads to an effective payment. In effect, one can use the same mechanism that I use in Section 5 to locate trusted transaction partners to pay for transactions (Swaminathan, 2010).

Example: assume that, in the following snippet of a trust network, A trusts B and B trusts D for medical services, which is shown in black, and A receives some medical services from D (see Figure 15). Similarly, D trusts C, and C trusts A for payments, which is shown in red, and A can use this path to pay for the services of D. A uses its own currency to pay C, which C accepts because it trusts A, and C uses its own currency to pay D, which D accepts because it trusts C. In effect, A made a payment to D through a trust path by using consecutive currency exchanges. Of course, the exchanges are executed automatically by the network without any manual intervention by the agents involved so long as the agents

are trusted to accept each other's currency and the pre-specified limit have not been reached. However, C and D end up holding some foreign currency at the end of this exchange, and those holdings need to be balanced out at some point, which I discuss later.

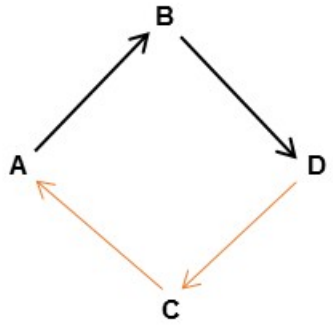

Figure 15. A Trust Network with Payments

Agents are incentivized to accept the currencies of their trusted partners because payments, just like transactions, are designed to provide a stream of income to all those involved in enabling them. As in searches, the system collects a commission from each payment and distributes it to all those that are on the trust path that enabled the payment. The actual incentives and the commission required to prevent default are very similar to the insurance system, derived by using exactly the same analysis, because accepting a trusted partner's currency is, in fact, a form of insurance. It is insurance that the issuer will honor the currency when it is presented back to it with goods, services, or an equivalent currency acceptable to the presenter. Each agent sets a limit on how much currency it will accept from each of its trusted partners, derived again by using the same analysis as insurance. The only new issue is what happens when the limit is reached and an agent has too much of the currency of a trusted partner and now has to balance the payments by returning that currency to the issuer in exchange for its own currency. The solution is straightforward when two agents trust each other, accept each other's currency, and the system periodically automatically exchanges the currencies and returns them to the original issuer as the agents reach their limits. So, if both A and B trust each other and at the end of a period each ends up with 10 dollars' worth of each other's currency, the system simply exchanges them and returns them to the issuer, so each agent ends up with no foreign currency as expected. However, such reciprocal trust is not always possible where both parties conveniently end up holding sufficient amounts of the right currency; and longer cycles have to be found to balance the currency holdings.

Example: consider the following snippet (see Figure 16) of a trust network in the payment context.

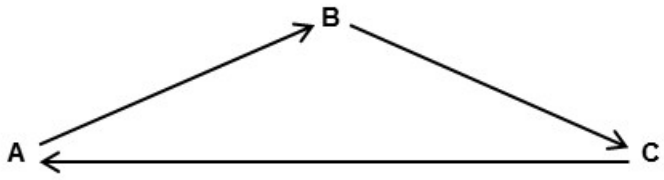

Figure 16. A Trust Network with a Cycle

A trusts B and accepts its currency; B trusts C and accepts its currency; and C trusts A and accepts its currency. When A reaches the limit of B's currency it can hold, the system will find the smallest cycle and move some of B's currency from A to B, move the corresponding amount of C's currency from B to C, and move the corresponding amount of A's currency from C to A. In effect, it replaces A's holding of B's currency with A's own currency by taking advantage of the cycle and moving equivalent amounts around the cycle.

Sometimes a cycle may not be synchronized, and the agents, although they accept each other's currency, may not have sufficient amounts to complete the cycle at a given point in time. This requires a more complicated strategy involving bisected cycles, and unsynchronized multi-step balancing of the payments.

Example: consider the following snippet of a trust network in the payment context (see Figure 17).

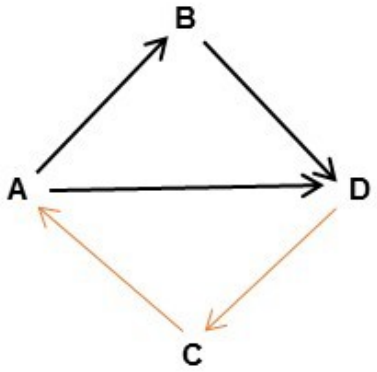

Figure 17. A Trust Network with a Bisected Cycle

This is called a bisected cycle since ABCD forms a cycle but AC bisects it into two separate cycles. Both start at the same node, yet one includes the other. In this case, the cycle ACD is included in the cycle ABCD. Such bisected cycles can be used to have unsynchronized multi-step balancing of payments. A has an excess of B's currency and reached its limit of holdings. The system will try to balance the payments by moving B's currency from A to B, an equivalent amount of C's currency from B to C, and an equivalent amount of C's currency from C to A. In effect, the system will replace A's holding of B's currency with C's currency, which A accepts. When A reaches the limit of its C's currency holdings, the system will use the ACD cycle as before to return all currencies to their original issuers. In effect, the system divided the payment balancing process into two unsynchronized steps by making A hold C's currency temporarily.

6.2 Application 2

The network can accommodate investments by one agent into another agent naturally by using private currency as a loan instrument. One agent accepting the private currency of another agent in payment for goods and services is simply an extension of credit to that agent to be paid later in goods and services or any other acceptable form of currency. The system automatically continues to extend credit by accepting the private currency of the trusted partner until the pre-specified limit is reached, and, at that point, it attempts to balance the payments or ceases to accept the oversupplied currency. All of that happens automatically with no banking system to control or underwrite the loans or a central bank to regulate the amount of money in the system. The incentive for accepting another agent's currency is simply the commissions earned for each payment that uses one's trust connections (see Section 3). The appropriate commission rate to discourage defections follows the same analysis as in Section 3. One can easily extend the peer-to peer currency exchange system to include equity investments in addition to loans. Both loans and equity investments are funds made available to a trusted partner by accepting its private currency. The distinction is in how commissions are assessed and paid. In loans, the commissions collected on every transaction are distributed only to those whose trust connections were used. In equity, the commissions collected on every transaction are distributed to all who have made an investment by committing to accept the currency regardless of whether their connections were used in that particular transaction or not. In effect, with loans, one gets paid on the amount they loaned as in an interest payment; with equity investment, one gets paid on all revenues as in dividend payments. The peer-to-peer currency system manages to integrate different forms of investing in the same framework.

6.3 Application 3

Hospitals can implement health insurance as its trust in its patients. The hospital invests (trusts) in its patients by providing health care, and they, in turn, pay a commission on all of their income to the hospital. In effect, the hospital invests in the health of its patients and has every incentive to watch out for their best interests since its income depends on the health and the future income of its patients. A hospital can invest (trust) in the health of a whole community to pool the risk, and a group of hospitals can collectively invest in all of their patients to pool their services. A variety of insurance schemes can all be implemented on top of a trust network by using its affiliated payment and investment mechanisms.

6.4 Application 4

Academic literature such as scholarly papers and books can be trusted like all other products and services and potentially replace the cumbersome and inefficient peer-review system. Scientific communities and journals can establish their own nodes on the network and trust their editors and reviewers. Then, all papers that are accessible from the journal node through a trust path would be trustworthy or publishable. One can even have a rating system by using my extended model with varying levels of trust, on the basis of the path length, or on the number of distinct paths. Academic publishing can be transformed into a peer-to-peer distributed evaluation system that bypasses the central bureaucracies of journals and scientific communities. In this environment, anybody can read and evaluate a scientific paper and choose to trust it or not. Trust paths starting at individuals establish personalized trust. Trust paths starting at journal or university nodes can be used to establish institutional trust. A university, for example, can trust various journals, scientific communities, or its own faculty, and any publications achievable from those nodes would be considered trustworthy and reputable for promotion and rating purposes.

6.5 Application 5

Politics is all about trust in those who represent specific political and economic interests and can be transformed through effective trust networks. Voting is a process of finding a candidate that one trusts to represent their interests, and it is a straightforward application of trust networks. Voting can even be completely automated in terms of trust connections since the existence of the shortest trust path to a political candidate can be interpreted as support and vote for that candidate. Some politically active and knowledgeable agents can trust various candidates directly similar to newspaper or politician endorsements; others can trust their local politicians, church or civic leaders, or newspapers and indirectly trust more distant candidates through their trust connections. Such multilevel delegation allows intelligent political participation and even political and social activism without every person studying and making judgments about every candidate and every issue, which, in effect, automates the representative democracy through trust networks (Orman, 2015).

7 Conclusions

Trust networks have the potential to combine the advantages of searches, recommendation systems, and social networks. If designed properly, they can be used to find trustworthy information, people, products, and services on public networks. But proper design and correct incentives are critical to the success of such networks.

In this paper, I present a trust network architecture that emphasizes simplicity and robustness. I propose a trust network with constrained trust relationships and designs a decentralized search and recommendation processes. I create both informational and monetary incentives to encourage individuals to join the network, investigate and discover other trustworthy agents, and make investments in them by trusting them, by insuring them, or even by directly investing in them. I show that making the correct judgments about others' trustworthiness and reporting it truthfully are the optimum strategies since they reward the agents both with information by providing access to more of the network and also with monetary payments by paying them for their services as information intermediaries. The extensive income potential from the trust connections creates strong incentives to join the network, to create reliable trust connections, and to report them truthfully.

There are also strong incentives against cheating. The trust network is a source of information and income for every agent, and the quality of information and the amount of income it receives are directly proportional to the quality and reliability of the trust connections it makes. Fake trust connections reduce the information quality and the income potential drastically and make them undesirable. Fake and multiple identities are not helpful to the perpetrators since the value of the network to a participant increases dramatically as the network connections improve in quality over time. Defecting on transactions after building a reputation is not desirable since the value of maintaining a solid reputation in terms of future income exceeds the short-term gain from defecting or cheating. The insurance mechanism further reduces the desirability of a defection since increasing insurance and trust an individual receives from others increases the individual's income dramatically both through products and services the individual provides but, more importantly, through the income the individual receives from others' using the individual's trust connections. The possibility of direct investment further reduces the benefits from

defection since receiving direct investments from others distributes financial gains and costs over many agents and reduces the benefits received from a single defection.

I analytically justify the design architecture I sketch out in this paper and tested it with a simulation. The simulation used real trust data from an existing trust network but relied on simulated transactions and payments on that network to validate the conclusions. I suggest researchers could conduct future research in commercial large-scale implementation of trust networks and empirically test the claims in real commercial networks to fully justify my conclusions.

References

Adomavicius, G., Tuzhilin, A., Zheng, R. (2011). REQUEST: A query language for customizing recommendations. *Information Systems Research, 22*(1), 99-117.

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach*. ACM Transactions on Information Systems, 23*(1), 103-145.

Andersen R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., & Tennenholtz, M. (2008). Trust-based recommendation systems: An axiomatic approach. *Proceedings of WWW Conference* (pp. 199-208).

Avery C., Resnick P., Zeckhauser R. (1999). The market for evaluations. *American Economic Review, 89*(3), 564-584

Ba, S. (2001). Establishing online trust through a community responsibility system. *Decision Support Systems*, *31*, 323-336.

Benbasat, I., Gefen, D., & Pavlou, P. A. (2010). Novel perspectives on trust in information systems. *MIS Quarterly, 34*(2), 367-371.

Borgatti, S. P., & Cross, R. (2002). A relational view of information seeking and learning in social networks. *Management Science, 49*(4), 432-445.

Braynov, S., Sandholm, T. (2002). Contracting with uncertain levels of trust. *Computational Intelligence, 18*(4), 501-514.

Conesa, J., Storey, V. C., Sugumaran, V. (2008). Improving web-query processing through semantic knowledge. *Data and Knowledge Engineering, 66*(1), 18-34.

Dellarocas, C. (2003). The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science, 49*(10), 1407-1424.

Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets.* Cambridge, UK: Cambridge University Press.

Feldman, M., Lai, K., Stoica, I., & Chuang, J. (2014). Robust incentive techniques for peer-to-peer networks. In *Proceedings of ACM Conference on Electronic Commerce.*

Fensel, D. (2001). *Ontologies: Silver bullet for knowledge management and electronic commerce.* Berlin, Heidelberg: Springer-Verlag.

Golbeck, J., & Hendler, J. (2006). Inferring trust relationships in Web based social networks. *ACM Transactions on Internet Technology, 6(*4), 497-529.

Golbeck, J. (2010). Trust and nuanced profile reliance in online social networks. *ACM Transactions on the World Wide Web*, *3*(4),

Guha, R., & Kumar, R. (2004). Propagation of trust and distrust. In *Proceedings of WWW Conference* (pp. 403-412).

Hancock, J. T., & Gonzales, A. (2015). To lie or not to lie online: The pragmatics of deception in computer-mediated communication. In S. Herring, D. Stein, & T. Mouton (Eds.), *Handbook of pragmatics of computer-mediated communication.* Berlin, Germany: Mouton de Gruyter.

Huang, J., & Nicol, D. M. (2010). An approach to formal semantics based calculus of trust. *IEEE Internet Computing*, *14*(5), 38-46.

Huang, J., & Fox, M. S. (2006). An ontology of trust: Formal semantics and transitivity. In *Proceedings ICEC Conference on Electronic Commerce* (pp. 259-270).

Josang, A., Ismail, R., & Boyd, C. A. (2007). Survey of trust and reputation systems for online service provision. *Decision Support Systems, 43*(2), 618-644.

Kuter, U., & Golbeck, J. (2010). Using probabilistic confidence models for trust inference in Web-based social networks. *ACM Transactions on Internet Technology*, *10(*2), 1-23.

Ma, N., Lim, E.-P., Nguyen, V.-A., Sun, A., & Liu, H. (2009). Trust relationship prediction using online product review data. In *Proceedings of CIKM Conference on Information and Knowledge Management* (pp. 47-54).

Malaga, R. (2001). Web-based reputation management systems: Problems and suggested solutions. *Electronic Commerce Research, 1*, 403-417.

Massa, P., Souren, K., Salvetti, M., & Tomasoni, D. (2008). Trustlet: Open research on trust metrics. *International Journal of Parallel and Distributed Computing*, *9*(4), 341-351.

McKnight, D. H., Choudhury, V., & Kacmar, C. (2002). Developing and validating trust measures for e-commerce: An integrative typology. *Information Systems Research, 13*(3), 334-359.

Middleton, S. E., Shadbolt, N. R., & Roure D. C. D. (2004). Ontological user profiling in recommender systems*. ACM Transactions on Information Systems, 22*, 54-58.

Moldoveanu, M. C., & Baum, J. A. C. (2011). "I think you think I think you're lying": The interactive epistemology of trust in social networks. *Management Science, 57*(2), 393-412.

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. In *Proceedings of IUI Conference*.

Orman, L. V. (2013). Bayesian inference in trust networks. *ACM Transactions on MIS*, *4*(2), 1-21.

Orman, L. (2015). Information paradox: Drowning in information, starving for knowledge. *IEEE Technology and Society*. Retrieved from http://ssrn.com/abstract=2620237

Pavlou, P. A., & Gefen, D. (2004). Building effective online marketplaces with institution-based trust. *Information Systems Research, 15(*1), 37-59.

Regan, K., Poupart, P., & Cohen, R. (2006). Bayesian reputation modeling in electronic marketplaces. In *Proceedings of AAAI Conference on Artificial Intelligence* (pp. 1206-1212).

Richardson, M., Agrawal, R., & Domingos, P. (2003). Trust management for the semantic Web. In *Proceedings of Semantic Web Conference*.

Söllner, M., Hoffmann, A., Hoffmann, H., Wacker, A., & Leimeister, J. M. (2012). Understanding the formation of trust in IT artifacts. In *Proceedings of the International Conference on Information Systems.*

Streitfeld, D. (2011). In a race to out-rave, 5-star Web reviews go for $5. *The New York Times.* Retrieved from http://www.nytimes.com/2011/08/20/technology/finding-fake-reviews-online.html

Swaminathan, A., Cattelan, R. G., Wexler, Y., Mathew, C. V., & Kirovski, D. (2010). Relating reputation and money in online markets. *ACM Transactions on the Web,* 4(4), 1-31.

Vance, A., Elie-Dit-Cosaque, C., & Straub, D. W. (2008). Examining trust in information technology artifacts: The effects of system quality and culture. *Journal of Management Information Systems, 24*(4), 73-100.

Vogiatzis, G., MacGillivray, I., & Chli, M. A. (2010). Probabilistic model for trust and reputation. In *Proceedings of AAMAS Conference on Autonomous Agents and Multi-agent Systems* (pp. 225-232).

Wang, W., & Benbasat, I. (2008). Attributions of trust in decision support technologies: A study of recommendation agents for e-commerce. *Journal of Management Information Systems, 24*(4), 249-273.

Wang, Y., & Vassileva, J. (2005). Bayesian network trust model in peer-to-peer networks. In *Proceedings of the Workshop on Deception, Fraud and Trust in Agent Societies* (pp. 23-34). Berlin: Springer-Verlag.

Xiong, L., Liu, L. (2004). PeerTrust: Supporting reputation-based trust for peer to peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering, 16*(7), 843-857.

Ziegler, C.-N. (2009). On propagating interpersonal trust in social networks. In J. Golbeck (Ed.), *Computing with social trust* (pp. 133-168). London: Springer.

About the Authors

Levent V. Orman specializes in Information Technology Management. He has taught courses and written papers on electronic commerce, database management, and social impact of technology. His papers have appeared in a variety of journals such as *Information Systems*, *ACM Transactions on MIS*, and *IEEE Transactions on Knowledge and Data Engineering*, *MIS Quarterly*, *Electronic Commerce Research*, and *IEEE Technology and Spciety*. He is the author of "Technology and Its Discontents: The Deadly Embrace of Technology and Society".