Trustworthy Service Composition: Challenges and Research Questions

Munindar P. Singh

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7535, USA
singh@ncsu.edu

Abstract. Web services have been gathering an increasing amount of attention lately. The *raison d'etre* of Web services is that we compose them to create new services. For Web services to be effectively composed, however, requires that they be trustworthy and in fact be trusted by their users and other collaborating services. In our conceptual scheme, principals interact as autonomous peers to provide services to one another. Trust is captured as a composite relationship between the trusted and the trusting principal. Principals help each other discover and locate trustworthy services and weed out untrustworthy players. The interactions of the principals combined with the needs of different applications induce interesting structures on the network. We apply multiagent systems techniques to model interactions among the principals.

By varying the requirements of different applications, the needs of different principals, the existence of special principals such as trusted authorities, and the mechanisms underlying the interactions, we draw attention to a variety of important settings where Web services would be composed. One, leading to superior methods through which trust can be evolved and managed in realistic service-composition settings. Two, studying the relationships between aspects of trust for Web services and the evolution of Web structure.

1 Introduction

The worldwide expansion of the Internet has quantitatively and qualitatively changed how people interact with one another. This has had a direct impact on the creation of applications such as electronic commerce, entertainment, and virtual communities. More interestingly, along with the rise of new applications, the very structure of computing architectures is being affected.

A new model of software development and composition is emerging. Where previously objects were linked to compose software systems, we now see the emergence of independent *services* that can be put together dynamically at run time and possibly across administrative domains. In essence, the Internet is transforming into the main connectivity fabric of upcoming service architectures. This new metaphor, which we term *service-oriented computing*, is a natural outgrowth of distributed object systems. Here the components are decidedly autonomous and long-lived. They cannot be invoked in the traditional sense, but must be engaged, i.e., requested to perform various actions.

For two reasons, service-oriented computing emphasizes the importance of trust between a service provider and service consumer. One, the implementation of a service is not available for inspection and may be changed by the provider. Two, a service generally executes in a different administrative domain than the consumer and may interact with other services and resources. Because of the separate administrative domain, certain low-level security techniques such as firewalls do not apply.

Thus trust, which is important wherever autonomous parties interact, is critical for service-oriented computing. Since services are becoming the major method for building software systems, it is essential that we develop approaches for trust that apply in this setting. Everyone who studies trust recognizes that it is a complex concept. However, from our distributed computing perspective, we can identify some elements of trust that cohere well with service-oriented computing. In simple terms, a service consumer will trust a service provider if the consumer has had prior good interactions with that provider or if the provider was referred to or endorsed by principals whom the consumer trusts.

Current directions in trust. Existing research on trust falls into the following main categories.

- Infrastructural, distributed trust techniques geared to ensure that the parties you are dealing with are properly authenticated or credentialed and that their actions are authorized under applicable policies. The policies would generally be constructed and enforced in an application-specific manner. But notice that even if someone is authenticated and authorized, there is no guarantee that they are acting in your best interests.
- Reputation mechanisms for tracking the past behavior of different parties with a view to identifying reliable parties with whom to interact. Reputation mechanisms seek to apply where the harder security techniques stop. Knowing that someone has satisfactory credentials does not assure you that they are the best or even an acceptable choice of a party to interact with, but the felicity of their interactions with others might indicate their trustworthiness. More importantly, it takes a lot of work to acquire a good reputation and a party who has built up a reputation generally won't risk it by purposefully misbehaving with another party.
- Policy concerns, especially dealing with security and privacy. In the case of privacy, these policies apply to the acquisition, storage, and dissemination of privileged information.

While all of the above themes of research are valuable and essential, current approaches fail to adequately address the challenges for trust in the emerging metaphor of service-oriented computing. The distributed trust techniques apply at a uniformly low level. That is, when you build a system, you can exploit mechanisms to disseminate and apply policies and credentials. Although the policies will vary across applications, the basic functioning of the trust mechanisms will not. Conversely, the reputation techniques apply at a uniformly high level. That is, when you build a system, the reputation mechanisms enable your components to make and access necessary ratings. Again, although some of the representations may vary across applications, the basic computations will not.

Whereas uniformity is desirable, being oblivious to the structure of the applications is a limitation of current approaches. Let's see how existing approaches might be combined into a strawman solution. Consider users who use graphics art services. With a conventional reputation mechanism, users might post their ratings of different graphics artists to a reputation server such as a better business bureau (BBB). Users would access the BBB to obtain credentials for graphics artists that they are considering. Now suppose that there is no BBB or, conversely, that there are several BBBs. Where should the user go to find a good graphics artist? If there are no central BBBs, how can the users help each other find good artists? How can good users be distinguished from those favor a particular artist because of side deals? Can some of the participants take on specialized tasks in their interactions with others? Can some be more helpful than others? What happens if some participants must be distinguished from others for regulatory reasons? How can varied artistic tastes of the users be taken into account? Notice how a large variety of scenarios can arise even in a toy example. Notice also that current approaches leave most of these variations to be dealt with by applications developers, which they will usually do in an ad hoc manner.

A new program of research. This paper doesn't offer any answers. Instead, it seeks to motivate a new program of research, which addresses challenges and research questions that arise when we take a systems-oriented, but high-level view of trust. The idea, ultimately, is to bridge the chasm between current practice in developing systems and the sophisticated insights of the modern research into trust that is centered around multiagent systems. The challenges motivated here squarely addresses the scientific and engineering foundations of trust. There is increasing interest in understanding these foundations, because of the obvious importance of constructing reliable systems.

The proposed program seeks to develop the concepts of trust from a services perspective. It seeks to develop techniques and methodologies through which important aspects of the implicit structure in service architectures and protocols can be represented and exploited. Thus, this effort will contribute at a level that overlays the current understanding of trust infrastructure, but which should ultimately be considered a part of the emerging infrastructure.

An obvious challenge is to compose trustworthy systems from potentially untrustworthy parts. However, the way in which distributed systems are being built is rapidly evolving into services-oriented computing. In this style, the services function as components that are dynamically composed to deliver a desired service. Individual service providers may be untrustworthy in different ways. Therefore, need approaches to dynamically compose trustworthy systems while employing personalized notions of trust. More generally, we must understand the interplay between trust and key features of service composition.

Organization. The rest of this paper is organized as follows. Section 2 motivates service-oriented computing, the special challenges it poses for our understanding of trust, and a technical framework for addressing these challenges in a unified manner. Section 3 describes the challenges that we will encounter in building trustworthy service-oriented systems. In doing so, it introduces some allied concepts to model interesting aspects of trust in service-oriented computing and the key technical research questions that must

be answered in our framework. Section 4 places our contributions in relation to the most relevant literature.

2 Motivation and Framework

The social and business impact of the networked society is unprecedented in all of history. The study of trust is becoming ever more crucial as the technologies for networking and applications involving electronic commerce and personal interaction are gaining currency.

The security and assurance of the electronic infrastructure is increasingly crucial. Current approaches for trust, because they are centralized and reliant exclusively on mechanisms such as digital certificates, are particularly vulnerable to attacks. This is because if some authorities who are trusted implicitly are compromised, then there is no other check in the system. By contrast, in a decentralized approach where the principals maintain trust in each other for more reasons than a single certificate, any "invaders" can cause limited harm before being detected.

Network architectures are evolving interesting hybrids of the two classical varieties: the Internet-style "stupid" network on the one hand and the telecommunications-style "intelligent" network on the other. These changes present both opportunities and challenges to address the longstanding problems of trust.

2.1 Service Composition

The services metaphor is catching on rapidly for the development of complex Web applications. Its business and technical motivations are excellent. Because of the heterogeneity and autonomy of web-sites, it is only natural that we model them as independent services. Services will facilitate superior solutions to be more easily constructed, thereby leading to new opportunities for businesses that can produce valuable services.

The first generation of the work on Web services has concentrated on basic infrastructural needs, such as directory services, description languages, and invocation standards. Relevant activities include Universal Description, Discovery and Integration (UDDI) [27], Web Services Description Language (WSDL) [7], and the Simple Object Access Protocol (SOAP) [3]. But the whole point of having Web services is that they be composed into more complex and more valuable services. Of course, the exploitation of the developer services by an end-user through a suitable user interface is important, but from the standpoint of engineering, exposing a service through a user interface is only a special case of composition. Present techniques that are popular within the Web community address the challenges of composition only to a limited extent. Some of the most important higher-level abstractions are not studied within the community; instead classical programming techniques are lifted for Web services. These techniques, such as remote procedure calls, were developed for traditional closed systems.

A lot more can and should be said about Web services, especially when we view them from the perspective of composition. For example, services in general are not invoked but are *engaged*, meaning that the interactions one has with them are quite unlike method invocations and are better modeled as parts of extended conversations. Unlike

method invocations, extended conversations preserve the autonomy of the participants and naturally lead to settings where more than two parties might participate.

Web services open up new business models that more clearly recognize the value derived from using the given software, e.g., by pricing it on a per-use basis. In particular, a small company that offers a critical component of a desired solution can compete on an even footing with larger competitors, because its component can be readily incorporated into the overall solution. Leading companies such as Microsoft, IBM, and Sun, which compete aggressively with one another agree on the importance of Web services, because the emerging interest in services creates business opportunities for selling platforms to provide new services. In addition, Microsoft is pursuing the .NET initiative, which also includes a role for Microsoft as a provider of a composite authentication and payment service. Sun and others are responding with competing standards. The present activities highlight the importance of Web services and the need for effective models of trust. They also highlight the limitations of current approaches in not addressing the challenges of achieving and managing trust in different settings.

2.2 Trust

Trust in general is a relational concept involving the trusted and the trusting parties. This point opposes the presently common assumption that trusted authorities exist independently of the other participants. Such authorities can exist only under rigidly constructed and administered computational environments. For example, on the eBay website, eBay is an authority who (with various caveats) authenticates the sellers in its auctions, maintains their ratings, and even warrants their good behavior. However, eBay would be unable to make similar guarantees to parties who weren't subject to its legal contracts covering bidding and selling at its auctions.

In general, a service interaction or negotiation can benefit from the existence of a trusted third party, but only if the protocols are such that the trusted party is somehow elected. How a party is chosen to be trusted in this manner may itself involve other instances of the application of trust.

For the purposes of engineering service-based solutions, it is natural that trust itself be offered as a service. What form should this service take? To answer this question, we need to probe further and develop a clearer computational framework in which to model service-oriented systems.

2.3 Referrals as a Unified Technical Framework

We now define the key concepts underlying trust in emerging computing environments. Our technical framework is intended to be simple yet flexible so that we can easily model the varieties of architectures and mechanisms that we believe are crucial to any investigation of trust. Our specific challenges and research questions will be formulated within this framework.

We model a trust system as consisting of *principals*, who are trustworthy or not for each other and who are potentially interested in knowing if other principals are trustworthy. The principals could be people or businesses. They provide *services* to each other. Our notion of services is general in that they need not be business services

provided for a fee, but may be volunteer services. They may even not be services in the traditional sense, e.g., participation in a network-based game. By the same token, *quality of service* includes not only the quality of the basic service but also any relevant ancillary features, such as privacy. That is, the quality of a service would generally be multidimensional—i.e., a vector rather than a scalar. For example, a graphics art service provider who produces a good picture layout for you, but doesn't protect your privacy may be treated as offering a good quality of service along some dimensions (say, esthetics) and a poor quality of service along some other dimensions (say, privacy).

The principals can keep track of each other's trustworthiness. To apply in settings where there are no universally trusted servers, we incorporate the idea of *referrals*. Referrals are common in distributed systems, e.g., in the domain name system (DNS), but are usually given and followed in a rigid manner. We capture a more flexible notion of referrals, reminiscent of referrals in human dealings. Importantly, by giving and taking referrals, principals can help one another find trustworthy parties with whom to interact.

The principals are autonomous. That is, we do not require that a principal respond to another principal by providing a service or referral. When they do respond, there are no set guarantees about the quality of the service or the suitability of a referral. However, constraints on autonomy, e.g., due to dependencies and obligations for reciprocity, are easily incorporated. Likewise, we do not assume that any principal should necessarily be trusted by others: a principal would unilaterally decide how much trust to place in others.

The above properties of principals match them ideally with the notion of *agents*. Agents are persistent computations that can perceive, reason, act, and communicate. Agents can represent different principals and mediate in their interactions. Principals are seen in the computational environment only through their agents. The agents can be thought of as assisting in the book-keeping necessary for a principal to track its ratings of other principals. Moreover, the agents can interact with one another to help their principal find trustworthy peers with whom to interact.

In abstract terms, the principals and agents act in accordance with the following protocol. Either when a principal desires a service or when its agent anticipates the need for a service, the agent begins to look for a trustworthy provider for the specified service. The agent queries some other agents from among its *neighbors*. A queried agent may offer its principal to perform the specified service or may give referrals to agents of other principals. The querying agent may accept a service offer, if any, and may pursue referrals, if any. Partly based on service ratings from its principal, an agent can learn about which neighbors to keep. Key factors include the quality of the service received from a given provider, and the resulting value that can be placed on a series of referrals that led to that provider. In other words, the referring agents can be (and usually should be) rated as well. An agent's own requests go to some of its neighbors. Likewise, an agent's referrals in response to requests by others are also given to some of its neighbors, if any match. This, in a nutshell, is our basic social mechanism for trust.

Together, the neighborhood relations among the agents induce the structure of the given society. In general, as described above, the structure is adapted through the decisions of the different agents. Although the decisions are autonomous, they are influenced by the mechanisms we have in place. The resulting structures could depend a

lot on the services offered by different principals, the demand for these services, any payment mechanisms in place, and so on.

3 Challenges and Research Questions

We now consider the key challenges that must be surmounted and the core research questions that must be addressed in order to engineer trustworthy systems in a principled manner. We proceed in a roughly bottom-up manner so that the initial challenges are of direct practical interest, whereas the latter challenges bring up conceptual questions which, however, will inform methods for engineering large systems and help place our results in the wider context of the study of Web systems.

3.1 Service Discovery

Unlike in a traditional distributed system, discovering the right service in an open system is more than a matter of simply looking up a directory with a specified method signature. This is because of two reasons. One, a trustworthy directory might not exist. That is, although directories may exist, the service consumer may not have a basis for trusting any of them. There is a huge question of scale for a directory to keep up with large numbers of services. Two, because whom to trust depends on the trusting party, finding a suitable trustworthy service might involve understanding the intended consumer of the given service.

Somehow, a service consumer must ensure that any service recommendations obtained are not based on ulterior motives, such as in the paid-placement search engines of today's Internet. Pure P2P systems (discussed below) wouldn't have any directories, but may have peers who take on specialized functions similar to directory servers. But even then, principals using such specialized peers have to establish that the specialist peers are indeed trustworthy.

Thus service discovery leads to the following challenges. How may service discovery proceed in settings where the existence of trustworthy directories cannot be assumed? How can the inherently multidimensional and relational aspects of trust be accommodated computationally? How can we cope with large numbers of services?

3.2 Service Evaluation

Because a Web page shows its contents vividly, it is possible to judge its quality. However, evaluating a service in general is difficult and depends on the class of application one is considering.

Consider again the e-commerce setting described above. Service consumers generally are able to judge the quality of the services provided by others. However, they might themselves never acquire the capability to offer the same service as the one they consume. For instance, you might never learn enough to provide an auto repair service yourself, yet you would be competent to judge if an auto mechanic did his job well. E-commerce contrasts with knowledge management. Very often, a consumer of "knowledge" might be unable to judge its quality, at least at the outset. However, over

time, the consumer might learn enough to become an independent provider. This is roughly how professors are trained.

The matter of evaluation leads to some interesting questions. What kinds of methods can take advantage of ready access to evaluations and what kind can avoid suffering from a lack of evaluations? Can delayed evaluations be accommodated? How much do delayed or poor evaluations affect the resulting trust relationships?

3.3 Protocols

Web services can be engaged through well-defined communication protocols. Protocols, in this sense, replace programming interfaces as an abstraction for programming, e.g., [8]. Composed services will typically interact via protocols, e.g., for negotiation or payment [26]. Protocols open up some interesting questions for us. How does the existence of different protocols influence the development of trust among principals? Does rigidity of protocols help or hinder trust? Can protocols be used to introduce trust among principals and later, when trust is established, be removed, so that the principals can proceed in an ad hoc manner with greater confidence in each other? Real-life protocols can be long-lived, lasting months in some cases. Can trust be easily maintained for equally long periods?

3.4 Architectures

Although our interest is in distributed systems in general, it is instructive to consider two emerging varieties of distributed architectures where services are obtaining technical and business attention, and where special challenges arise for trust.

Peer-to-peer (P2P) computing refers to a class of architectures where the different components or nodes are equals of one another. The definitions of P2P computing vary in the technical community, but it is clear that P2P computing is expanding into the realm of large-scale computations over the Internet [25]. Key examples include Gnutella [13] and Freenet [9]. P2P systems promise a new paradigm for distributed computing in the large. Although present-generation P2P systems are used for simple applications such as file exchange, the true power of the P2P architecture will arise in more general settings, where the peers can be seen as providing services to one another. Their openness implies that there would be few regulatory restrictions for ensuring that the services offered are of a suitable quality or that the peers discovered over the network are trustworthy.

The study of distributed architectures brings up the following challenges. Can we develop techniques to achieve and maintain trust that are generic and yet flexibly able to specialize to the given architectural variation where they are applied?

3.5 Topology

Different application classes induce different link topologies on the referral networks. Current modeling approaches capture the aggregate structure of the Web. However, viewed from the standpoint of services, additional structure emerges. Some principals

may be primarily service providers, others service consumers, and still others repositories of referral information. Different application assumptions will affect the nature of the principals and the links between them.

For example, in a typical e-commerce setting, the service providers are distinct from the service consumers. Customers connect to other customers to get referrals and to service providers to obtain services. These links essentially form paths that lead service customers to service providers with different expertise. Typically, the service providers do not have outgoing links, because they neither initiate queries nor give referrals.

A simpler topology arises in a knowledge management setting. Here the principals are closer to being symmetric in that each can provide a service (knowledge) and each can consume it. However, the principals will vary in the extents of their knowledge and in the usefulness of the referrals they give (the usefulness of referrals being captured in terms of leading to trustworthy sources). Also, the topology that evolves in such a setting will depend on how the knowledge offerings and the knowledge needs of the principals relate. Another interesting topology arises in a content networking setting, where the principals play three logical roles: sources of content, cachers and forwarders of content, and consumers of content. The trustworthiness of a content source will incorporate its perceived quality, timeliness, frequency of updates, and so on.

Given a starting topology, how does it influence the chances of a particular principal being identified as trustworthy or not? And with how much efficiency (in terms of the number of interactions or messages exchanged)? Conversely, given merely the needs of different roles of principals, what kind of a topology will emerge under different profiles of trustworthiness?

3.6 Clustering

Let us consider the important role of the referring principals. There are two main variants reflecting two competing doctrines, of intimacy with the provider or the consumer, respectively.

- *Authority*. The referring principal is considered authoritative in identifying good providers in the given domain.
- *Familiarity*. The referring principal is considered to be familiar with the needs of the consumer.

In computing, trustworthiness is conventionally associated with authority. However, in real-life, trustworthiness is just as often a matter of familiarity. You wouldn't necessarily trust what is considered the universally best service provider, but one who is closely linked to yourself: directly or through others that are close to you. For example, if you are a student for travel to India, you may not wish to deal with the large US travel agencies such as Expedia or Orbitz, but with mom-and-pop travel agent who specializes in the particular region of India that you plan to visit and who caters to students like yourself.

An obvious question is whether authority or familiarity is more superior in terms of producing more trustworthy providers with less effort. We conjecture that the answer will vary with the application topology we choose especially with regard to the distribution of the services offered and needed by various principals. This bears significantly

on the important matter of clustering. It is generally believed that similar principals will cluster together because they can recommend useful services to one another. Many deployed recommendation systems, which are based on collaborative filtering, function in this manner. Roughly, they cluster users to predict the needs of a given user based on the clusters he falls in. However, a case can be made that in a referral network, principals who cluster with similar principals might not gain much but lose out on the capabilities of principals who are dissimilar to themselves. Intuitively, clustering supports the effect of familiarity and opposes the effect of authority.

We conjecture that if arcane, narrowly-focused services are desired, familiarity might be superior, whereas if diverse but popular services are desired, authority might be superior. Subtle formal representations of services might be required to capture these distinctions. If so, an additional question is how a system can be designed to evolve the right behavior regardless of of the profiles of the consumers.

3.7 Web Structure

Links across Web pages induce a structure on the Web. It is convenient to assume that these links indicate some sort of an endorsement relationship, leading to the PageRank heuristic employed by Google [4].

Another interesting study of the structure of the Web structure comes from the work on small-world models of the Web. Small-world networks are graphs that are neither fully regular nor fully random, but capture the structure of real-life human organizations [29]. Watts and Strogatz observe that such graphs have both clusters (like regular graphs) and short paths (like random graphs). They have the nice property that they tend to have small diameters, leading to improved connectivity among the vertices.

It is widely recognized that the distribution of links on the Web obeys the *power law*. Specifically, the number of pages with k incoming links is inversely proportional to k^m; Albert *et al.* estimate that $m = 2.9$ [1].

What kinds of structures would be induced by links that indicate service composition or implied evaluations of trustworthiness? How do these structures depend on the application domain, underlying mechanisms such as for payment or reciprocity, individual variations in the trustfulness or trustworthiness of different principals? How is trust affected by specific families of distributions, such as the power-law distributions? Conversely, how does adapting in light of trust induce such distributions? Further, are small-world networks desirable for trust networks? Can they be evolved through local learning by agents in various schemes?

4 Discussion and Comparisons

We now consider how the proposed program of research relates to previous computational approaches for trust. We review the main practical and theoretical approaches on trust. Next we briefly consider how the questions we raise might be addressed cohesively.

4.1 Literature

Some of the key techniques that apply in service composition were developed in the areas of databases, distributed computing, artificial intelligence, and multiagent systems. These are generally established bodies of work that can be readily adapted for service composition. Some additional techniques, although inspired by these areas, must be developed from scratch, because they address the essential openness and scale of Web applications that previous work did not need to address. Both classes of key techniques should be incorporated into our best practices for service design and composition. In many cases, they can be applied on top of the existing approaches.

Trust in multiagent systems. There has been much work on social abstractions for agents, e.g., [5, 11]. The initial work on this theme studied various kinds of relationships among agents. Some studies of the aggregate behavior of social systems are relevant. More recent work on these themes has begun to look at problems of deception and fraud. Castelfranchi and Falcone argue that trust means depending upon another agent to ensure the success of whatever one is doing [6]. That is, the extent of your trust in another party is the extent to which you place your plans in its hands. To ensure that our results apply in general computing environments, we do not emphasize planning in the proposed program. However, we do capture protocols to be able to represent the logical dependencies among the actions of different principals.

Mamdani and Pitt study the delegation of authority to agents and ensuring that they remain accountable to their masters and their masters remain accountable to society [17]. They raise the concern that checking compliance of complex software is difficult and if someone is to trust an agent to act on his behalf, he must have some assurance that the agent will work responsibly. Mamdani and Pitt outline some important challenges in developing systems that police agents, recover from errors, and so on.

Previous work on protocols has tended to hard-wire specific assumptions about how much the various participants should trust each other. These protocols require rigid sequences of actions. Consequently, they become an obstacle to the development of flexible trust methods. We recently developed a representation for protocols wherein an agent can vary its actions to suit its constraints, including its level of trust in another party [31].

Distributed trust. A recent survey of trust mechanisms from a distributed computing and communications standpoint is available in [14]. In distributed computing, trust management refers to the task of applying policies to ensure that the given principal has the requisite credentials to be authorized to perform certain, potentially risky, actions [2]. Trust management involves continually evaluating (depending on the policies) the authorizations to ensure that a principal won't act in violation of some stated constraint.

Another interesting body of research concerns dealing with trust with respect to mobile code, e.g., [30]. Wilhelm *et al.* consider how a principal may evaluate another principal's policies in terms of adequacy before permitting any code originating from the second principal's domain or certified by the second principal to execute locally. At one level, passing requests as messages to services is less risky than permitting mobile code to execute in one's administrative domain. However, it is no less essential to trust

the service provider than the originator of any mobile code. As Wilhelm *et al.* observe, while the adequacy of a policy can be computationally determined, the trustworthiness of a principal cannot be formalized. Thus they favor a pessimistic approach that prevents certain risky actions.

Rea and Skevington propose trusted third parties (TTP) as a bridge between buyers and sellers in electronic marketplaces [22]. However, this is most appropriate for closed marketplaces. In open systems, a TTP may either not be available or have limited power to enforce good behavior. TTPs would become a special case of a principal who is effectively elected a mediator by other principals.

Reputation mechanisms. Kasbah is a good prototype [32]. It requires that principals give a rating for themselves and either have a central agency (direct ratings) or other trusted principals (collaborative ratings). A central system keeps track of the principals' explicit ratings of each other, and uses these ratings to compute a person's overall reputation or reputation with respect to a specific principal. These systems require pre-existing social relationships among the principals of their online community. It is not clear how to establish such relationships and how the ratings propagate through this community.

Rasmusson and Janson proposed the notion of *soft security* based on social control through reputation [21]. In soft security, the agents police themselves, and no central authority is needed. However, Rasmusson and Janson don't analyze the propagation of trust in a purely autonomous setting.

Marsh presents a formalization of the concept of trust [18]. His formalization considers only an agent's own experiences and doesn't involve any social mechanisms. Hence, a group of agents cannot collectively build up a reputation for others. Schillo and Funk's social interaction framework (SIF) provides a method to evaluate the reputation of another agent based on direct observations as well through other witnesses [24]. But SIF does not describe how to find such witnesses, which limits the practicality of this approach.

Referral networks. These are a natural way for people to go about seeking information [20]. One reason to believe that referral systems would be useful is that referral capture the manner in which people normally help each other find trustworthy authorities.

The importance of referrals to interpersonal relationships has long been known [10] as has their usefulness in marketing, essentially as a method for service location [23]. The earliest agent-based referral system that we know of is MINDS, which was based on the documents used by each user [19]. ReferralWeb is based on the co-occurrence of names on WWW pages [16]. Kautz *et al.* model social networks statically as graphs and study some properties of these graphs, e.g., how the accuracy of a referral to a specified individual relates to the distance of the referrer from that individual.

The proposed program of research considers referrals as the primary mechanism through which principals can help each other.

Web structure. We discussed some important lines of research on this topic above. Gibson *et al.* discuss an approach to infer Web communities from the topology of links among Web pages [12]. Communities here are defined in terms of related sets of *hubs*,

which ideally point at lots of authorities, and *authorities*, which are ideally pointed to be lots of hubs. The main difference between previous work and our approach is that our model is inherently heterogeneous, whereas previous work treats all pages as essentially alike. Also, Web pages are vivid in that what you see is what you get, whereas services in general leave a lot of room for confusion and misunderstanding, thus increasing the importance of trust. In this sense, our work generalizes over the previous research. It would be interesting to see how the algorithms, such as of Gibson *et al.*, can be be extended to apply in our model.

4.2 Toward a Cohesive Research Program

Section 3 identified a number of interesting aspects of realistic service-oriented systems and which have an intuitive relationship with trust. This variety is the main reason why the line of research we motivate here is challenging and interesting. Although these are several aspects and each offers its own unique research questions, we suggest that these questions be studied in a uniform manner. This is crucial, because it not only makes the desired effort tractable, but also ensures that these will form a cohesive program of research, whose results will be nicely synthesized into principles of wide applicability.

For this purpose, it is encouraging to note that, although quite simple, the referrals-based framework introduced in Section 2.3 is rich enough to model some of the interesting subtleties of service-oriented systems. Modeling these subtleties would enable us to address some interesting questions about the relationships between trust and various important properties of systems of service consumers and providers.

Our proposed framework involves agents participating in multiagent systems. Traditionally, research on multiagent systems has followed an artificial intelligence perspective, but the need for applying multiagent systems on trust in distributed systems opens up research questions that are more directly studied in an interdisciplinary manner.

Acknowledgments

I am indebted to several colleagues and students for useful discussions, in particular, Mike Huhns, Bin Yu, and Pınar Yolum. This was partially supported by the National Science Foundation under grant ITR-0081742.

References

1. Réka Albert, Hawoong Jeong, and Albert-László Barabási. Diameter of the world-wide web. *Nature*, 401:130–131, September 1999.
2. Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. The role of trust management in distributed systems security. In *[28]*, pages 185–210. 1999.
3. Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. Simple object access protocol (SOAP) 1.1, 2000. www.w3.org/TR/SOAP.
4. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

5. Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *Proceedings of the International Conference on Multiagent Systems*, pages 41–48, 1995.
6. Cristiano Castelfranchi and Rino Falcone. Principles of trust for MAS: cognitive anatomy, social importance, and quantification. In *Proceedings of the 3rd International Conference on Multiagent Systems*, pages 72–79, 1998.
7. Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web services description language (WSDL) 1.1, 2001. www.w3.org/TR/wsdl.
8. Mark d'Inverno, David Kinny, and Michael Luck. Interaction protocols in Agentis. In *Proceedings of the 3rd International Conference on Multiagent Systems (ICMAS)*, pages 112–119. IEEE Computer Society Press, July 1998.
9. Freenet. Home page, 2001. http://freenet.sourceforge.net.
10. Noah E. Friedkin. Information flow through strong and weak ties in intraorganizational social network. *Social Networks*, 3:273–285, 1982.
11. Les Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. In *[15]*, pages 389–404. 1998. (Reprinted from *Artificial Intelligence, 1991*).
12. David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring Web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space - Structure in Hypermedia Systems*, pages 225–234. ACM, 1999.
13. Gnutella. Home page, 2001. http://gnutella.wego.com.
14. Tyrone Grandison and Morris Sloman. A survey of trust in Internet applications. *IEEE Communications Surveys and Tutorials*, 3(4):2–16, December 2000.
15. Michael N. Huhns and Munindar P. Singh, editors. *Readings in Agents*. Morgan Kaufmann, San Francisco, 1998.
16. Henry Kautz, Bart Selman, and Mehul Shah. ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, March 1997.
17. Ebrahim (Abe) Mamdani and Jeremy Pitt. Responsible agent behavior: A distributed computing perspective. *IEEE Internet Computing*, 4(5):27–31, September 2000.
18. Steven P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computing Science and Mathematics, University of Stirling, April 1994.
19. Uttam Mukhopadhyay, Larry Stephens, Michael Huhns, and Ronald Bonnell. An intelligent system for document retrieval in distributed office environments. *Journal of American Society for Information Sciences*, 37:123–135, 1986.
20. Bonnie A. Nardi, Steve Whittaker, and Heinrich Schwarz. It's not what you know, it's who you know: work in the information age. *First Monday*, 5(5), May 2000.
21. Lars Rasmusson and Sverker Janson. Simulated social control for secure Internet commerce. In *Proceedings of the Workshop on New Security Paradigms*, pages 18–25, 1996.
22. Tim Rea and Peter Skevington. Engendering trust in electronic commerce. *British Telecommunications Engineering*, 17(3):150–157, 1998.
23. Peter H. Reingen and Jerome B. Kernan. Analysis of referral networks in marketing: Methods and illustration. *Journal of Marketing Research*, 23:370–378, November 1986.
24. Michael Schillo and Petra Funk. Who can you trust: Dealing with deception. In *Proceedings of the Autonomous Agents Workshop on Deception, Fraud and Trust in Agent Societies*, pages 95–106, 1999.
25. Munindar P. Singh. Peering at peer-to-peer computing. *IEEE Internet Computing*, 5(1):4–5, January 2001. Instance of the column *Being Interactive*.
26. Marvin A. Sirbu. Credits and debits on the Internet. In *[15]*, pages 299–305. 1998. (Reprinted from *IEEE Spectrum, 1997*).
27. UDDI technical white paper, 2000. www.uddi.org/pubs/Iru-UDDI-Technical-White-Paper.pdf.

28. Jan Vitek and Christian D. Jensen, editors. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1603 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1999.
29. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.
30. Uwe G. Wilhelm, Sebastian M. Staamann, and Levente Buttyán. A pessimistic approach to trust in mobile agent platforms. *IEEE Internet Computing*, 4(5):40–48, September 2000.
31. Pınar Yolum and Munindar P. Singh. Commitment machines. In *Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL-01)*. Springer-Verlag, 2002. In press.
32. Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. *Decision Support Systems*, 29(4):371–388, December 2000.