A Survey on Trust-Based Web Service Provision Approaches

Nicola Dragoni
Department of Informatics and Mathematical Modelling
Technical University of Denmark
ndra@imm.dtu.dk

***Abstract*—The basic tenet of Service-Oriented Computing (SOC) is the possibility of building distributed applications on the Web by using Web Services as fundamental building blocks. The proliferation of such services is considered the second wave of evolution in the Internet age, moving the Web from a collection of pages to a collections of services. Consensus is growing that this Web Service "revolution" won't eventuate until we resolve trust-related issues. Indeed, the intrinsic openness of the SOC vision makes crucial to locate useful services and recognize them as *trustworthy*. In this paper we review the field of trust-based Web Service selection, providing a structured classification of current approaches and highlighting the main limitations of each class and of the overall field. As a result, we claim that a *soft* notion of trust lies behind such weaknesses and we advocate the need of a new approach based on a stronger (*semantics-based*) notion of trust.**

***Keywords*-Web Services; Trust; Service Provision;**

I. Introduction

Service-Oriented Computing (SOC) [1] is an emerging paradigm for distributed computing aiming at changing the way software applications are designed, delivered and consumed. The key tenet of SOC is the possibility of building distributed applications on the Web by using Web Services (WS) as fundamental building blocks. The proliferation of Web Services is considered the second wave of evolution in the Internet age, moving the Web from a collection of pages to a collections of services used by software agents.

In order to realize this vision and to bring SOC to its full potential, several challenges must still be addressed. In particular, consensus is growing that this Web Service "revolution" won't eventuate until we resolve trust-related issues. For instance, lack of consumer trust still represents a critical impediment to the success of WS-based marketplaces [2], [3]. In terms of trust, the key point to be addressed concerns the problem of trust-based Web Service selection. Indeed, the focus of current WS techniques is mostly on describing, composing and discovering services according to their functional aspects (*what* a service can do). But in a large, open and dynamic SOC system where anyone can publish his own services, a client faces a dilemma in having to make a choice from a bunch of services offering the same functionalities. In other words, selecting the *right* service does not include only the problem of discovering services on the basis of their functionalities, but also the one of evaluating *how* well a service can work. This evaluation must be computed according to non-functional quality of service (QoS) aspects. Since the underlying assumption of the SOC vision is that discovered Web Services are not known a priori by the user, the evaluation of trust becomes a key aspect of WS selection. The intrinsic openness of the SOC vision makes crucial to locate useful services and recognize them as trustworthy.

Paper Contribution and Outline. In this paper we present an overview of the field of trust-based WS selection. The contributions are threefold. First, we provide a structured classification of all the approaches according to their rationale, so that approaches belonging to a specific class differ only for minor (mostly technical) aspects (Sec. II). Then, for each class we discuss the underlying fundamental idea and we list the various weaknesses with respect to the trust-based Web Service selection problem (Sec. III-VI). Finally, we highlight the key limitations of the state of the art and we claim that a *soft* notion of trust lies behind such weaknesses. As a result, we advocate the need of a new approach based on a stronger (semantics-based) notion of trust. The paper extends the preliminary work presented in [4].

II. Trust-Based WS Selection Approaches

The rapidly growing literature on the theory and applications of trust-based systems for Web Service provision confirms the key importance of this problem in the SOC vision. In this paper we do not aim to list and compare the whole "jungle" of works on this topic. This is primarily motivated by the fact that current approaches can be classified into few classes according to their rationale. Works belonging to a specific class share the same fundamental idea and differ only for minor and mostly technical issues. According to this point of view, we aim at clarifying the matter by making a rationale-based classification of all the approaches. For each class, we then cite some representative papers and we highlight its main limitations.

The mentioned classification is shown in Fig. 1. Three main classes of approaches can be identified in literature: (i) approaches based on the *direct past experience* of the consumer with the service (service's confidence); (ii) approaches based on a *Trusted Third Party* that provides the assessment of a service in place of the consumer; (iii) *Hybrid approaches* that combine techniques from the previous two

classes. In the classification we also include automated Trust Negotiation (TN), although in literature this technology has not been directly targeted to our problem. TN aims at establishing trust relationships among two parties (service consumer and provider) so that both can trust each other if the negotiation succeeds. Therefore, TN addresses a more general problem (based on a notion of *mutual trust*) that includes the trust-based WS selection problem.

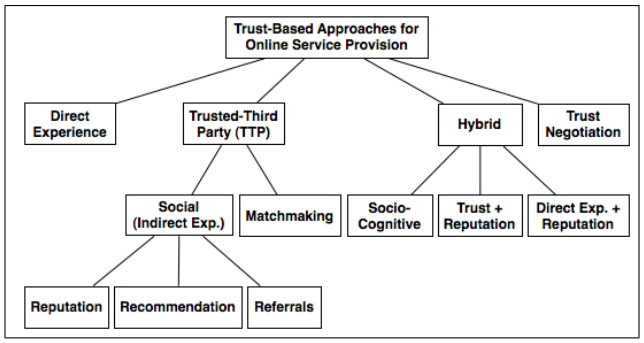

Figure 1. Classification of Approaches for Trust-Based WS Provision

III. Direct Experience-Based Approaches

These approaches (such as [5], [6]) are based on *presumptions drawn from the service consumer's own direct experience with the target service* [7]. The rationale is that trust is computed as a rating of the level of performance of the party (service or service provider). The party's performance is assessed over multiple interactions checking how good and consistent it is at doing what it says it will.

Def. 3.1 (Trust by Direct Experience): A service consumer trusts a service because of his good past experience with the service.

Limitations: The approach is not suitable for open systems where anyone can publish its (malicious) code, since it does not allow to trust a service *before* its execution.

IV. Trusted Third-Party (TTP) Approaches

The rationale behind TTP approaches is that there exists a trusted third party that service consumers can consult to trust a service. Consumers will take the final decision according to the results provided by this third party. The third party could be a central authority or a distributed system composed of several "members" that form a *community*. The underlying assumption of these approaches is that consumers must trust the third party they decide to consult.

We distinguish among two types of approaches: *social* and *matchmaking approaches*. In both approaches, the final decision is based on the assessments provided by the TTP. The difference lies in *how* the assessments are computed.

A. Social Trust (or Trust by Indirect Experience)

The TTP computes the service assessment according to the evaluations of the users registered in the system. To be effective, the approach requires that each service consumer acts not only as a service user but also as a reviewer, continually evaluating the performance of services and service providers. The TTP will be responsible to collect and aggregate all the evaluations related to a service. Three different social-based approaches can be identified in literature: *reputations*, *recommendations* and *referrals*.

1) Reputation: A *reputation* [8] can be seen as the general's opinion about the character or standing (such as honesty, capability, reliability, ...) of an entity (users, service and service providers). A reputation is *objective* and represents a collective evaluation of an entity based on the ratings from members in a community. The reputation system is responsible to collect ratings from members in the community and to compute and publish global reputation scores about entities, so that all the members in the community will see the same reputation score for a particular entity. This score is then used by a consumer when deciding whether or not to select a particular service. A commercial example adopting such approach is eBay (https://eBay.com).

Feedbacks from consumers are usually related to several kinds of data acquired from executing a Web Service (*i.e.,* execution time, response time). In [9] a classification of QoS metrics for Web Services that might be used by a consumer to rate a service and by a reputation system to collect and combine services' ratings is provided. The overall global service's reputation score will depend on such combination.

In summary, the rationale of reputation systems is that an individual's subjective trust on a service is derived from the reputation of that service or, in other words, from the *direct experience of someone else*.

Def. 4.1 (Trust by Reputation): A service consumer trusts a service because of its good reputation.

Limitations: Current approaches for reputation-based online service selection suffer from several shortcomings. The most critical one is based on the rationale of the reputation approach: *to establish trust among unknown parties one party relies on past information from other members of the community*. A natural problem arises in case of new services. For example, when a service initially registers for business, no other consumer has interacted with it and consequently no information exists of the service past behavior. In this situation, consumers can not assess its reputation and questions about its trustworthiness are left unanswered. Consequently, new research efforts for reputation-based online service provision are needed. Mechanisms assigning reputation for newly deployed services should be defined to make the approach effective in the SOC vision. These mechanisms must provide reputation scores even when no historical information about the behavior of a service exists. Only in this way newly published services can compete with existing services for market share. A representative approach in this direction is [10], where the authors propose a reputation-bootstrapping method based on the concept of community. The basic idea is that Web Services in a particular domain

(*i.e.,* registered within the same community) can aid each other in assessing a newcomer's initial reputation. Unfortunately, the approach has several limitations, for instance that a single bootstrapping mechanism can not be universally adopted and different bootstrapping techniques must suit different domains or conditions. Moreover, the approach is strongly based on the cooperation among the (rational and not malicious) members of an existing community, which looks like a too strong assumption for open large systems.

Another shortcoming is that reputation systems are mostly centralized and no convincing *real-life* distributed approaches have been proposed. As remarked in [1], in centralized architectures the central authority is responsible for (i) authenticating the users, (ii) recording, aggregating and revealing ratings, (iii) owning ratings. Such authorities can exist only under rigidly constructed and administered computational environments. Two notable exceptions are the EigenTrust and the PeerTrust systems [11], [12], that might represent the two most important and cited examples of distributed reputation-based trust management systems. But again, at the best of the author's knowledge, only proof-of-concept systems or simulations have been proposed. The design and implementation of *real-life* distributed reputation systems is still an open challenging issue.

Other technical limitations reside in: (i) the possible alteration of the ratings (collusion or retaliation); (ii) in the fact that users of ratings do not know the parties who provided the ratings; (iii) the fact that the effectiveness of any reputation system relies on the number of members in a community and on their behavior. In particular, the fewer the number of people participating in a reputation system, the more inadequate the ratings provided by the systems [13].

2) Recommendation: Recommendation systems [14], [15], [16], [17], [18], [19], [20] aim at making a prediction of a consumer's needs or interests. In its common formulation [16], the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a consumer. Intuitively, this estimation is usually based on the ratings given by this consumer to other items or on the ratings that *similar* users provided for the targeted items. Once it is possible to estimate ratings for the yet unrated items, then the system can recommend to the user the items with the highest estimated ratings.

Def. 4.2 (Trust by Recommendation): An user trusts a service because of some recommendations got from a trusted authority.

Recommender systems can be classified into three categories, according to how recommendations are computed [15]: in *Content-Based Filtering* a user is recommended items similar to the ones the user preferred in the past; in *Collaborative Filtering* a user is recommended items that people with similar tastes/preferences liked in the past; *Hybrid Approaches* combine the previous two methods.

Content-based filtering is a static approach for selecting items by filtering Web sites in terms of the words occurring in them. For instance, it could be applied to services by indexing their text descriptions based on the words that occur in them. But this approach would be primitive and would be a step backward from current Web Services standards (that involve formal structured service descriptions). For this reason, we focus our analysis on Collaborative Filtering (CF) which represents the most widely adopted recommender method, for instance in e-commerce sites such as Amazon[1].

In CF, user's ratings for items are stored centrally and these ratings are often simply captured as the products a given user purchased [1]. If two users rate a set of items similarly, they share similar tastes and for this reason they are *neighbors*. This information can then be used by the CF system to recommend items that one participant likes to his or her neighbors. In other words, a user is given recommendations based on the ratings by other users who are similar to the given user, that is who have similar *subjective tastes*. Informally speaking, if Alice and Bob both bought movies A, B and C and Alice bought also movie D, then a CF system may recommend that BOB also buy D.

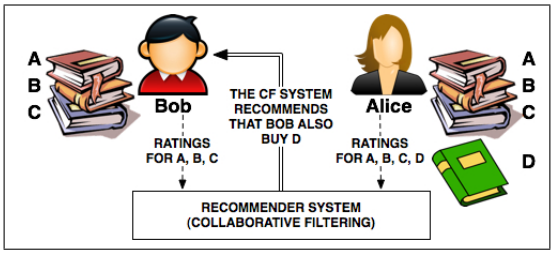

Figure 2. Basic idea of Collaborative Filtering

The implicit assumption underlying CF systems is that different people have different tastes and rate things differently according to subjective taste. This represents a key difference with respect to reputation systems [21], which are based on the seemingly opposite assumption: all members in a community should judge the quality of a product or service consistently. In other words, CF takes ratings subject to taste as input, whereas reputation systems take ratings assumed insensitive to taste as input.

Limitations: While recommender systems seem working well in e-commerce sites to buy products, there are some limitations of applying them to the WS selection problem [1]. First, the assumption that someone purchased a service does not mean that they liked it. Then, in the SOC vision services are distributed and advertised by a registry/broker. This entity does not provide the service it is recommending and may have little to say about its trustworthiness. For example, a registry would not have any control on the actual service interaction, while an e-commerce site would know that a product was shipped correctly.

[1]http://amazon.com

Other similar technical weaknesses could be listed, but as for reputation systems the key limitation still lies in the rationale of the approach: *CF relies on the existence and good working of a community that provides ratings to the centralized recommender system*. In open SOC environments these assumptions are too strong, leaving a consumer to a vulnerable position in case he does not belong to any community or the community is so poor that does not provide a significant rating system.

Finally, recommender systems are conceptually centralized and the same weaknesses discussed for centralized reputation systems are still valid.

3) Referrals (or Software Agent-Based Approach): A common weakness of recommendation and reputation mechanisms lies in their being conceptually and implementationally centralized: a single authority is responsible to collect, aggregate and present all the ratings. To address this limitation, referrals [22], [23] have been proposed as a decentralized approach based on *online communities* and *software agents* technologies.

An *online community* is a set of interacting members representing people, businesses or other organizations. Members provide services as well as referrals for services to each other. Referrals may be provided proactively or in response to requests. This is realized by means of *software agents* that assist members helping them manage their interactions. Software agents are persistent computations that can perceive, reason, act, and communicate [22]. Agents represent different members and assist them in evaluating services and referrals provided by others, maintaining contact lists, and deciding or suggesting whom to contact for different services. In this way, agents help their members in finding the most useful and reliable parties to deal with.

Referrals are based on a representation of how much the other available parties can be trusted [22]. Agents are responsible to build and manage these representations taking into account the previous experiences of their members and communicate with each others. Participating on behalf of different members, agents appear as autonomous and heterogeneous. Moreover, agents organize themselves into communities and agents in the same community are called *neighbors*. Communities are dynamically formed according to the model that each agent maintains of some other agents. This model is usually based on two aspects: the *party's expertise* (ability to provide correct services) and *sociability* (ability to produce accurate referrals).

Fig. 3 shows how a referral system could work for selecting an online service. Agent A sends a request of information about who provides a specific service to its neighbors B, C and D. Agent C autonomously decides to ignore the request and it does not reply. Instead, agents B and D answer to A's request but in two different ways. Indeed, in referral systems an answer can be a referral to another member (as D's answer) or even oneself (as B's answer), in which case there would be some more interaction to actually provide the service. According to D's referral, A decides to forward the query to E too. Again, E could reply with some referrals or proposing itself. A will take its final decision reasoning on the received answers.

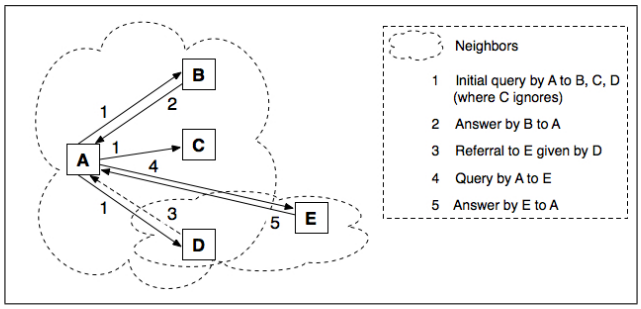

Figure 3. Example of Referral System for Online Service Selection

Def. 4.3 (Trust by Referrals): A service consumer trusts a service because of some referrals got from known trusted software agents.

Note the difference with Definition 4.2. Recommendation systems hide the identity of the sources of the recommendations that they aggregate. On the contrary, in referral systems the participants reveal their ratings to those whom they trust, so the ratings would be more likely to be honest.

Limitations: Referral systems address some limitations of reputation and recommendation systems (such as, their centralized nature) but still rely on the judgements of the members of a community. Here the community is formed by software agents that acts on behalf of their members (people, businesses, ...). Therefore, the effectiveness and practicability of the approach resides in the efficiency of the interacting community. Some technical practical issues, such as agents/members registration and communication as well as referrals representation, are left unanswered in the literature, making the impression of a still immature (or at least just academic) approach. A part from these technical questions, the approach is still based on ratings coming from the direct previous experience of someone else, which leads to the main problem of selecting trustworthy services in the absence of some neighbors that might help us.

B. Matchmaking-Based Trust

These approaches are based on a component called "matchmaker" responsible to match a user's request and trust preferences with available online service descriptions. If some matches are found than the results are sent back to the user. As shown in Fig. 4, two different matchmaking architectures have been proposed in literature, depending on the centralized or distributed nature of the matchmaker.

An implemented centralized trust-based matchmaking system has been presented in [24], where the authors embodied the WS selection problem in a *classification problem*: given a set of user and WS policies and established a

classification criterion, the goal is to identify a class of services matching with trust policies of involved users. In other words, services are classified according to the specific user as well as trust policies. To do this, they develop an ontology, namely Web Services Trust Ontology (WSTO), that is able to represent generic trust specifications within the semantic WS-based interaction context. Being based on the Web Service Modelling Ontology (WSMO), WSTO can be supported by the IRS-III platform [25], which in this context behaves as centralized trusted third-party storing both user's profiles and services and reasoning on them (Fig. 4-A).

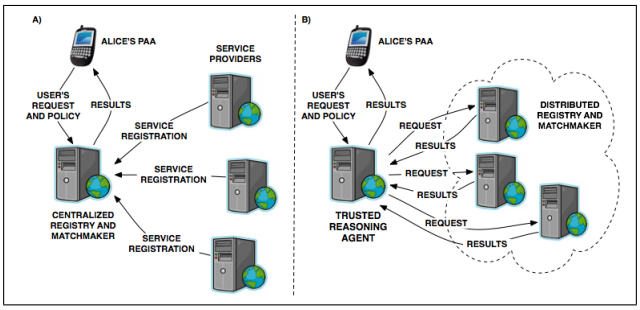

Figure 4. Centralized/Distributed Matchmaking Systems for WS Selection

A similar approach has been proposed by Olmedilla et al. in [26]. The main difference with respect to [24] lies in the underlying registry and matchmaking architecture, which is based on a P2P network (Fig. 4-B). Whenever a new service provider wants to offer its services, it must join such network. On the client side, a user looking for a service must send a query together with his policies to a trusted reasoning agent. The agent distributes the query to the peers and each peer applies a matching algorithm. Whenever a peer has matches, it sends them back to the reasoning agent which joins the results and present them to the user.

Def. 4.4 (Trust by Matchmaking): A service consumer trusts a service because a trusted matchmaker states that the service's policy matches the consumer's request.

Limitations: As already discussed, a centralized architecture such as [24] is not suitable for real open service oriented environments where the number of users and services might be very high. Having a single central matchmaker where both users and services must registered in and where all the matches are computed is very far from the SOC vision. Moreover, to work correctly the approach requires that users disclose all their policies when they register in the matchmaking system, since no trust negotiation is supported. This is in contrast with the openness of the system that would require a user to carefully discloses his/her policies. A consequence of this requirement is that the matching algorithm is not flexible and it is only based on a "take it or leave it" philosophy. Finally, it is not realistic to ask service providers to disclose all their (maybe very sensitive) policies to a centralized registry (even if it is trusted).

Olmedilla et al. [26] replace the centralized matchmaker/registry with a P2P network, distributing the matchmaking process to the service providers. This improves the performance and scalability of the matching algorithm, that might be computationally expensive to be executed on a single central server. Moreover, in a distributed approach servers can keep policies locally and private, which is an essential property in realistic open environments. However, the approach is based on a *trusted* reasoning agent that acts as intermediary between users and the network, moving the problem from trusting a service or a WS provider to the one of finding such a trusted agent. The authors point out that "different groups of users might use different trusted agents, *i.e.,* an university might set up an agent for its students and professors while a company could use a different one". Relying on such computational entities, the approach is not tailored to select trustworthy online services in SOC environments, because one cannot assume that these trusted agents will be always available for any context and service.

V. Hybrid Approaches

Hybrid approaches are based on a combination of well known trust methodologies, such as the ones addresses in previous Sections. The key idea is that by combining two different methodologies the resulting integrated framework improves some weaknesses of the constituent methodologies, and thus the overall assessment of online services.

A. Socio-Cognitive (or Beliefs-Based Approaches)

These approaches are mostly based on the works of Falcone and Castelfranchi [27], [28], [29]. Influenced by the Artificial Intelligence (AI) field and especially by the Autonomous Agents and Multi-Agent Systems (AAMAS) paradigm, they treat trust as an agent's mental state. In this view, trust is articulated as an assumption or an expectation that a service consumer makes about a specific service. This expectation is based upon more specific *beliefs* which form the basis or the components of trust.

Beliefs can be seen as the answers to the question "What do we have in mind when we trust a service?" [7] . For example, we may trust a service because we believe that the service is able to do what we need (competence belief), and it will actually do it quickly (promptness belief). Competence and promptness are examples of such "mental ingredients", or beliefs, of trust. A belief describes therefore *a state of the world from the point of view of an agent*. That is, it represents the state the agent has in mind for a service: which/how is the agent's trust in (evaluation of) the service as for its competence and ability? Which/how is the agent's trust in (evaluation of) the service as for its intention and reliability? And so on. Examples of beliefs proposed in literature [27], [28], [29], [7] include *competence (or reliability) belief* (the service's raw ability to accomplish a task, such as providing accurate results or performing a

desired action), *availability belief* (the availability of the service), *promptness belief* (the speed at which the service responds to task requests by accomplishing the agreed upon task) and *cost belief* (cost refers to the monetary value that the consumer is willing to pay.

Def. 5.1 (Socio-Cognitive Trust): A service consumer trusts a service because of some of its subjective beliefs.

Limitations: Since trust is a function of subjective beliefs, the approach requires the ability to form coherent beliefs about different characteristics of services and reasoning about these beliefs. A key question that arises is where and how such beliefs are obtained, that is from which *sources*. The answer to this question differentiates the various proposals in literature. The most common sources of beliefs are: *direct experience* (Section III), *reputation* (Section IV-A), *categorization* (the process of grouping things based on prototypes) and *reasoning* (the act of using reason to derive a conclusion from certain premises). For instance, [5], [30] propose models based on the direct interaction (experience) or reputation as sources. In [29] sources are categorization and reasoning, in [7] direct experience and reputation. In consequence, a first weakness of the approach lies in the fact that it is based on beliefs obtained by means of well know and still problematic methodologies. In other words, we are moving the problem of selecting a trustworthy service to the one of selecting trustworthy beliefs that will be used as reasoning basis for deciding on the service trustworthiness.

Another major limitation lies at the implementation level. To fully realize this approach, some sort of BDI[2] agents [31] is needed. Indeed, as Falcone et al. remarks in their paper [27]: "only a cognitive agent can trust another agent. We mean: only an agent endowed with goals and beliefs." This requirement seems too strong when applied to open and large service-based systems, since it is not reasonable to assume that every agent will be conformed to the BDI model (which, a part from the modeling of trust, requires specific architectures to support the reasoning on beliefs and goals). For instance, this model is far to be completely accepted in the AAMAS community too.

B. Trust and Reputation

Approaches like [32], [33], [34] propose methods for assessing the quality of online services by combining trust and reputation techniques in a single framework. For instance, [34] discusses how (Bayesian) reputation systems can be combined with trust modeling based on subjective logic [35].

Limitations: Although these approaches are remarkable, especially [34] where the integration results in a flexible framework for online trust management, they still suffer the main limitations of their constituent methodologies. For instance, both approaches inherit one of the main weaknesses of reputation system, that is to be based on a centralized and trusted reputation center (Section III).

[2]Belief-Desire-Intention

C. Direct Experience and Reputation

Some approaches (for instance [36], [37], [38]) propose a model where trust is computed as a rating of the level of performance of a service. This overall performance is not limited to the agent's direct experience (Section III) but it also based on the evaluations provided by the other agents in the system (in [36] called the "group experience" , *i.e.,* what the other members of the group think about the agent being evaluated and his group). Thus, in these models trust can be seen as a rating built as a result from combining agent's direct experience (with the service) with the social reputation of the service provider.

Limitations: Again, the combination of two methodologies improve some weaknesses of one constituent model, but it does not provide a complete solution to the problem. For instance, in [37] the authors combine confidence and reputation to address the situation where no previous experience of the service is available (main weakness of the direct experience method). But to do this they based their proposal on trust and reputation mechanisms to infer expectations of future providers' behavior from past experiences in similar situations. This idea inherits the already discussed problems of trust and reputation mechanisms.

VI. Automated Trust Negotiation

Automated Trust Negotiation (TN) [39] is an approach specifically targeted to allow agents to access sensitive data and services in open environments. TN protocols are based on the iterative disclosure of digital credentials and requests for credentials between two unknown parties (*strangers* in TN jargon), with the goal of establishing sufficient mutual trust so that the parties can complete a transaction. Note the difference between TN and the approaches we have seen so far. Here the point of view is not restricted to the service consumer only (how the service consumer may trust a service) but the goal is to establish a *mutual trust* between service consumers and providers.

Informally, digital credentials (credentials for short) refer to the online analogues of paper credentials (a drivers license, an employee ID card, etc...). Thus, a credential is a digitally signed assertion by a credential issuer about the credential owner. It is usually signed using the issuers private key and verified using the issuers public key [40].

To automate trust negotiation, a party must establish *access control policies* (policies for short) to protect its sensitive resources (*i.e.,* credentials and services) from inappropriate access. Each policy should specify the credentials strangers must present to access the protected resource. Policies can themselves be seen as sensitive resources.

Def. 6.1 (Trust by Credential-Based Negotiation): A service consumer and a service provider mutually trust each if a trust negotiation among them ends successfully.

Note that the above definition does not state that a negotiation will always succeed if the parties' access control

policies are compliant. Indeed, the success of the negotiation might depend on several factors. For instance, a negotiation could take different routes according to the strategies adopted by the parties [41]. In other words, compliant parties' access control policies is a necessary but not sufficient condition for ensuring the success of a negotiation.

Limitations: TN principles and systems have been widely investigated in the last few years, both in different domains (like eCommerce, P2P systems and more recently in Web Services [42]) and with respect to issues such as privacy, safety and efficiency. This effort is evident in the growing literature on TN related issues ([40], [43], [44], [45], [46], [26], [47], [39], [48], [49] to mention only a few). However, several key challenges have still to be addressed to widely and successfully adopt the TN approach. Listing all these weaknesses is outside the scope of the paper and in the following we will try to identify only the ones that are relevant to the online service selection problem.

Lack of Real-World TN Systems. To date, TN research has been primarily of a *theoretical and academic nature*, resulting in a strong theoretical foundation of the matter but developing only few proof of concept prototypes ([40], [50], [51]. To cope with real-world TN systems, a number of important technology-related issues must be addressed and to date no standards have been identified. For instance, one can find many languages for expressing resource access policies (e.g., [52], [53], [54]), several protocols and strategies for conducting trust negotiation (e.g., [43], [50], [51], [42]) and different logics for reasoning about the outcomes of these negotiations (e.g., [55], [56]). As a result, prototypes are based on different languages and protocols, making the different systems unable to talk to each other. Real-world TN systems are still missing.

Tailored to Credentials-based Negotiation. Also assuming that some standards will be eventually defined, exploiting a TN approach for selecting Web Services requires that both parties (client and service provider) are able to support a (complex) negotiation process. This sounds a too strong requirement for open large systems, where consumers should be able to select a trustworthy service with less computational effort and not necessarily after a (complex) negotiation (especially in the case of mobile devices with resource constraints). Moreover, the TN approach assumes that both parties interact according to a credential-based notion of trust. Other trust semantics are not supported.

Tailored to Single Service. Current TN approaches take for granted that *a client always starts the negotiation by requesting access to* a *resource*. Instead, as pointed out in [57], "interacting with real world Web Services involves generally a sequence of invocations of several of their operations, referred to as conversation". It is therefore of key importance to consider the access control and negotiation issues for the overall conversation. As noted in [41], it might well be that a conversation takes different routes, therefore changing the set of needed credentials. While [57] takes full care of the conversational aspect of Web Services, the related negotiation protocol still sticks to the progressive disclosure of credentials while keeping the set of requested services fixed. What is missing is a typical feature of *real-life* negotiations: *we are usually willing to trade off disclosure of our security attributes for (additional) services.* A first preliminary work on this direction has been proposed in [42].

Conclusion and Future Directions

In this paper, we have summarized the state of the art in the field of trust-based Web Service selection. The result of the review is that, although the growing literature, automatically evaluating the trustworthiness of Web Services still remains on open challenge that requires further investigation. Indeed, with the only exception of Trust Negotiation, all the approaches have two *key limitations* in common: trust is based (i) on the *direct experience of the user* with the service or (ii) on the *feedbacks provided by a trusted third party*, *i.e.,* according to the ratings coming from someone *trusted by the user*. In the second case, the vast majority of approaches computes trust on the basis of the *direct experience* of the trusted party. Thus, the rationale is that trust is drawn by first experiencing the service and then sharing this experience with the other members of a community. In consequence, *if someone does not take the risk of invoking an unknown service for the first time, then no one will be able to decide about the trustworthiness of the service before its invocation*.

According to this conclusion, we claim that current approaches are unsatisfactory because based on a notion of *soft trust*, as soft security was coined in [58]. Soft trust has a *social control* philosophy: participants in a market collaborate each other in sharing information on services. Soft trust expect and even accept that there might be malicious services or service providers in the system. The idea is to identify them and prevent them from harming the other participants by means of collaboration and social interactions, aiming at sharing as much knowledge as possible.

The key lack of soft trust is that *no service semantics is considered in the trust-based Web Service selection process*. That is, services are not selected according to their *security behavior*, *i.e.,* according to the relevant security features of the service (for instance, access control rules and QoS features). What is still missing is a stronger notion of trust relying only on the behavior of a service, instead of being based on the judgements coming from trusted third parties. Indeed, the SOC vision requires technologies enabling users to trust a service *before its invocation* without necessarily requiring the existence of a trusted community that experienced and evaluated the service in the past.

We therefore advocate the need of a *semantics-based approach* ensuring *hard trust*: services should be selected and trusted according to their security features, as well as they are (semantically) discovered according to their interfaces or semantic descriptions. The recent Security-By-Contract (SxC) approach [42] might represent a good starting point for this purpose, because it takes into account the security behavior of a service instead of depending on the social control philosophy in the existing trust based approaches. The combination of the SxC idea with some existing trust based approach might give promising results because it would join the benefits of a semantics based approach with the "collective intelligence" provided by social-based approaches.

REFERENCES

[1] M. P. Singh and M. N. Huhns, *Service-Oriented Computing*. WILEY, 2005.

[2] S. Dayal, H. Landesberg, and M. Zeisser, "Building trust on-line," McKinsey Quarterly (October 2001) http://www.mckinseyquarterly.com.

[3] C. Liu, J. Marchewkaa, J. Lub, and C. Yub, "Beyond concern: a privacy–trust–behavioral intention model of electronic commerce," *Information & Management*, vol. 42, no. 1, pp. 127–142, 2004.

[4] N. Dragoni, "Toward trustworthy web services - approaches, weaknesses and trust-by-contract framework," in *Proc. of WI-IAT*. IEEE, 2009, pp. 599–606.

[5] C. Jonker and J. Treur, "Formal Analysis of Models for the Dynamics of Trust Based on Experiences," in *Proc. of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*. Springer-Verlag, 1999, pp. 221–231.

[6] M. Witkowski, A. Artikis, and J. Pitt, "Experiments in building experiential trust in a society of objective-trust based agents," in *Proc. of the workshop on Deception, Fraud, and Trust in Agent Societies*. London, UK: Springer-Verlag, 2000, pp. 111–132.

[7] A. S. Ali, S. A. Ludwig, and O. F. Rana, "A cognitive trust-based approach for web service discovery and selection," in *Proc. of ECOWS*. IEEE, 2005.

[8] Y. Wang and J. Vassileva, "A review on trust and reputation for web service selection," in *Proc. of ICDCS*. Washington, DC, USA: IEEE, 2007.

[9] K. Lee, J. Jeon, W. Lee, S. Jeong, and S. Park, "QoS for Web Services: Requirements and Possible Approaches," W3C Consortium Note, November 2003.

[10] Z. Malik and A. Bouguettaya, "Reputation bootstrapping for trust establishment among web services," *Internet Computing*, vol. 13, no. 1, pp. 40–47, 2009.

[11] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: ACM, 2003, pp. 640–651.

[12] L. Xiong and L. Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 7, pp. 843–857, 2004.

[13] F. Hussain and T. Dillon, "Trust issues in service oriented environment," in *Proc. of SOLI*. IEEE, 2006, pp. 790 – 793.

[14] A. Abdul-Rahman and S. Hailes, "Using recommendations for managing trust in distributed systems," in *Intern. Conf. on Communication*. IEEE, 1997.

[15] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Comm.of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[16] G. Adomavicius and E. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE TKDE*, vol. 17, pp. 734–749, 2005.

[17] P. Bedi, H. Kaur, and S. Marwaha, "Trust based recommender system for the semantic web," in *Proc. of IJCAI*, 2007, pp. 2677–2682.

[18] S. Guan, X. Dong, W. Wu, Y. Mei, and S. Liao, "Trust Management and Service Selection in Pervasive Computing Environments," in *Proc. of CIS*. IEEE, 2007, pp. 620–623.

[19] L. Zhengping, L. Xiaoli, W. Guoqing, Y. Min, and Z. Fan, "A formal framework for trust management of service-oriented systems," in *Proc. of SOCA*. IEEE, 2007, pp. 241–248.

[20] S. Ahamed, M. Haque, and N. Talukder, "Service sharing with trust in pervasive environment: now it's time to break the jinx," in *Proc. of SAC*. ACM Press, 2008, pp. 1622–1628.

[21] A. Josang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[22] M. Singh, B. Yu, and M. Venkatraman, "Community-based service location," *Comm. of the ACM*, vol. 44, no. 4, pp. 49–54, 2001.

[23] P. Yolum and M. Singh, "Engineering Self-Organizing Referral Networks for Trustworthy Service Selection," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 35, no. 3, pp. 396–407, 2005.

[24] S. Galizia, A. Gugliotta, and J. Dominque, "A trust based methodology for web service selection," in *Proc. of ICSC*. IEEE, 2007, pp. 193–200.

[25] L. Cabral, J. Domingue, S. Galizia, A. Gugliotta, V. Tanasescu, C. Pedrinaci, and B. Norton, "IRS-III: A Broker for Semantic Web Services based Applications," in *Proce. of ISWC*, ser. LNCS. Springer-Verlag, 2006, pp. 201–214.

[26] D. Olmedilla, R. Lara, A. Polleres, and H. Lausen, "Trust negotiation for semantic web services," in *Proc. of Intern. Workshop on Semantic Web Services and Web Process Composition*, ser. LNCS. Springer-Verlag, 2004, pp. 81–95.

[27] R. Falcone and C. Castelfranchi, "Principles of trust for mas: cognitive anatomy, social importance,and quantification," in *Proc. of ICMAS*. IEEE, 1998, pp. 72–79.

[28] ——, *Trust and Deception in Virtual Societies*. Kluwer Academic Publishers, 2001, ch. Social Trust: A Cognitive Approach, pp. 55–90.

[29] R. Falcone, G. Pezzulo, and C. Castelfranchi, *Special Issue on "Trust, Reputation and Security: Theories and Practice"*, ser. LNAI. Springer-Verlag, 2003, ch. A fuzzy approach to a belief-based trust computation, pp. 73–86.

[30] M. Schillo, P. Funk, and M. Rovatsos, "Who you can trust: Dealing with deception," in *Proc. of "Deception, Fraud and Trust" Workshop of the Auton. Agents Conf.*, R. Falcone, Ed., 1999.

[31] A. S. Rao and M. P. Georgeff, "Bdi agents: From theory to practice," in *Proc. of ICMAS*. AAAI Press, 1995.

[32] L.-H. Vu, M. Hauswirth, and K. Aberer, "QoS-based Service Selection and Ranking with Trust and Reputation Management," in *Proc. of CIS*, ser. LNCS. Springer-Verlag, 2005.

[33] T. Huynh, N. Jennings, and N. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *JAAMAS*, vol. 13, no. 2, pp. 119–154, 2006.

[34] A. Josang, T. Bhuiyan, Y. Xu, and C. Cox, "Combining Trust and Reputation Management for Web-Based Services," in *Proc. of TrustBus*, ser. LNCS. Springer-Verlag, 2008, pp. 90–99.

[35] A. Josang, "A logic for uncertain probabilities," *Intern. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pp. 279–311, 2001.

[36] J. Sabater and C. Sierra, "Regret: a reputation model for gregarious societies," in *Proc. of AAMAS*, 2002.

[37] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings, "A computational trust model for multi-agent interactions based on confidence and reputation," in *Proc. of Deception, Fraud and Trust in Agent Societies Workshop*, 2003, pp. 69–75.

[38] H. Billhardt, R. Hermoso, S. Ossowski, and R. Centeno, "Trust-based service provider selection in open environments," in *Proc. of SAC*. ACM Press, 2007, pp. 1375–1380.

[39] E. Bertino, E. Ferrari, and A. Squicciarini, "Trust Negotiations: Concepts, Systems, and Languages," *Computing in Science and Engineering*, vol. 6, no. 4, pp. 27–34, 2004.

[40] M. Winslett, T. Yu, K. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu, "Negotiating Trust on the Web," *IEEE Internet Computing*, vol. 6, no. 6, pp. 30–37, 2002.

[41] H. Koshutanski and F. Massacci, "An access control framework for business processes for web services," in *XMLSEC*. ACM Press, 2003, pp. 15–24.

[42] N. Dragoni and F. Massacci, "Security-by-contract for web services," in *Proc. of SWS*. ACM Press, 2007, pp. 90–98.

[43] T. Yu, M. Winslett, and K. E. Seamons, "Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation," *ACM TISSEC*, vol. 6, no. 1, pp. 1–42, 2003.

[44] S. Ye, F. Makedon, and J. Ford, "Collaborative automated trust negotiation in peer-to-peer systems," in *Proc. of P2P*. IEEE, 2004, pp. 108–115.

[45] T. Leithead, W. Nejdl, D. Olmedilla, K. E. Seamons, M. Winslett, T. Yu, and C. C. Zhang, "How to exploit ontologies for trust negotiation," in *Proc. of ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*, vol. 127. CEUR-WS.org, 2004.

[46] W. Nejdl, D. Olmedilla, and M. Winslett, "Peertrust: Automated trust negotiation for peers on the semantic web," in *Proc. of SDM*, vol. 3178. Springer, 2004, pp. 118–132.

[47] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in *Proc. of ACM CCS*. ACM Press, 2005, pp. 46–57.

[48] A. Squicciarini, E. Bertino, E. Ferrari, F. Paci, and B. Thuraisingham, "Pp-trust-x: A system for privacy preserving trust negotiations," *ACM TISSEC*, vol. 10, no. 3, p. 12, 2007.

[49] A. Squicciarini, A. Trombetta, and E. Bertino, "Supporting robust and secure interactions in open domains through recovery of trust negotiations," in *ICDCS*. IEEE, 2007.

[50] E. Bertino, E. Ferrari, and A. C. Squicciarini, "Trust-x: A peer-to-peer framework for trust establishment," *TKDE*, vol. 16, no. 7, pp. 827–842, 2004.

[51] H. Koshutanski and F. Massacci, "Interactive access control for web services," in *Proc. of IFIP ISC*. Kluwer Press, 2004, pp. 151–166.

[52] N. Li and J. C. Mitchell, "Design of a role-based trust management framework," in *Proc. of IEEE SSP*. IEEE, 2002, pp. 114–130.

[53] E. Bertino, E. Ferrari, and A. Squicciarini, "X -tnl: An xml-based language for trust negotiations," in *Proc. of POLICY*. IEEE, 2003.

[54] M. Becker and P. Sewell, "Cassandra: Distributed access control policies with tunable expressiveness," in *Proc. of POLICY*. IEEE, 2004, pp. 159–168.

[55] P. Bonatti and P. Samarati, "A unified framework for regulating access and information release on the web," *Journal of Computer Security*, vol. 10, no. 3, pp. 241–272, 2002.

[56] M. Winslett, C. C. Zhang, and P. A. Bonatti, "Peeraccess: a logic for distributed authorization," in *Proc. of ACM CCS*. ACM Press, 2005, pp. 168–179.

[57] M. Mecella, M. Ouzzani, F. Paci, and E. Bertino, "Access control enforcement for conversation-based web services." in *Proc. of WWW*. ACM Press, 2006, pp. 257–266.

[58] L. Rasmusson and S. Jansson, "Simulated social control for secure internet commerce," in *Proc. of NSPW*. ACM Press, 1996, pp. 18–25.