Combining Cognitive with Computational Trust Reasoning

Eugen Staab and Thomas Engel

Faculty of Science, Technology and Communication
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
{eugen.staab,thomas.engel}@uni.lu

Abstract. We propose a concept that combines the *cognitive* with the *computational* approaches to experience-based trust reasoning. We emphasize that a cognitive component is vital for computationally modeling trust. At the same time, we recognize the predictive nature of trust. This suggests a combination of the different approaches. The idea is to introduce a cognitive component that produces factual positive and factual negative experiences. These experiences can then be used in a predictive component to estimate the future behavior of an agent. The components are combined in a modular way which allows the replacement of them independently. It further facilitates the integration of already existing trust update algorithms. In this work, we analyze the chain of trust processing for the concept step by step. This results in a concise survey on challenges for experience-based trust models.

1 Introduction

Much research on trust studies models that are based on direct experiences: *Positive experiences* lead to an increase whereas *negative experiences* lead to a decrease of trust in the respective agent. This so called *experience-based* trust reasoning is required in open distributed systems where no centralized third party can verify an agent's trustworthiness, and therefore agents need to reason about the trustworthiness of potential interaction partners on their own.

The concept for trust mechanisms that is presented in our work builds a bridge between two different approaches to experience-based trust reasoning. On the one hand there is the cognitive approach which was mainly promoted by Castelfranchi and Falcone [1,2,3]. In their approach, the trustor tries to develop a "theory of the mind" of the trustee in order to reason about how trustworthy he will behave in future interactions. On the other hand there is the computational approach. Here, the trustor uses experiences to derive numerical values that shall represent the trustworthiness of the trustee. This process involves in the majority of cases probabilistic mechanisms [4,5,6,7] or formulas that are mainly based on intuition [8,9,10,11].

The motivation for a fusion of the cognitive and the computational approach to trust is to seek for a model that incorporates the profundity of the cognitive

R. Falcone et al. (Eds.): TRUST 2008, LNAI 5396, pp. 99–111, 2008.

approach but uses at the same time mathematically well founded mechanisms for extrapolating the experiences into the future. In our concept, the *interpretation of observations* gets a much higher importance than it is usual in trust reasoning. This allows us to draw the cognitive part into the component for interpretation. Consequently, the components for cognitive reasoning on one side and computational mechanisms on the other side can be strictly separated, which in turn allows a modular structure. This modular concept enables an agent to tune and replace the components of our concept independently.

The model proposed in this work can serve as a starting point for developing sophisticated experience-based trust models. It further motivates a more exhaustive and more structured way of using "witness-information", i.e. evidence about an agent's trustworthiness that is acquired from other agents.

In Sect. 2, we analyze each component of the proposed concept in detail by surveying the challenges it faces. Following this, we discuss in Sect. 3 how witness-information can be integrated into the concept. Finally, we conclude the work in Sect. 4.

2 Modular Concept

Our concept divides the process of trust reasoning into four subprocesses or components (see also Figure 1):

1. **Monitoring:** In a first step, the performance of the trustee and his environment is monitored. This process results in a set of observations.
2. **Interpretation:** The observations obtained in step 1 are analyzed and interpreted as positive or negative experiences. As we will argue in the respective section, this component requires cognitive reasoning in order to understand causal relationships.

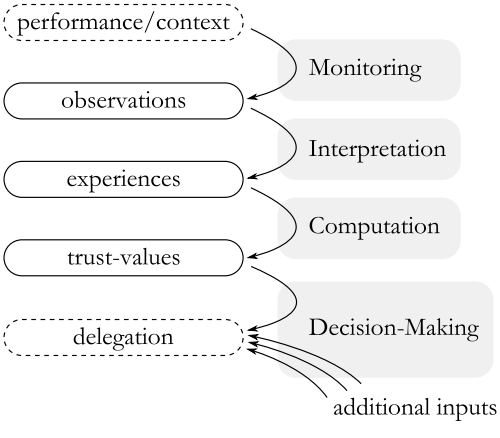

Fig. 1. Modular concept for trust reasoning

3. **Computation:** The set of experiences resulting from step 2 is used as input for a computational (e.g. probabilistic) trust update algorithm.
4. **Decision-Making:** Finally, the trust in potential interaction partners can be used as *one* criterion in decision-making. Here, it is decided whether a task should be delegated to some agent, and if yes, to which agent.

In the following, we explain and investigate the four components in greater detail.

2.1 Step 1 – Monitoring

The first step in making experiences is to *monitor* an agent's performance (or the outcome of it) and the environment within which the agent performs. Monitoring though is domain-dependent and therefore cannot be examined here in detail. We only want to mention the most common challenge for this process which can be called *distortion* or *interference*. To give an example, in Mobile Ad Hoc Networks (MANETs) [12], nodes can try to overhear whether their neighbors successfully forward packets they were asked to forward; packets coming from other neighbors can collide with the possibly forwarded packets, which makes monitoring more difficult.

Even though the process of monitoring is not specific to the problem of trust reasoning, trust models must be aware of errors that can occur during this step. Also, if different agents use different sensors for monitoring, the same facts can be perceived in different ways. This is especially important when exchanging witness-information (see Sect. 3).

2.2 Step 2 – Interpretation

The monitoring of an agent's performance results in information that is neutral and thus requires an *interpretation*. Through the process of interpretation, an agent decides whether he thinks about the outcome of an interaction as positive or negative experience. Note that this is an important part of subjectivity within the trust reasoning process as it is up to the interpreting agent to define what positive or negative is. Now, trust is used to decide about the selection of future interaction partners. Thus, only a performance that provides evidence for positive future interactions should be interpreted as positive experience; and accordingly, only a performance that provides evidence for negative future interactions should be interpreted as negative experience.

There are many ways to carry out an interpretation. A common approach found in trust modeling is to compare an agent's actual performance to the expected performance [5,8,13]. The expected performance can be described in form of a so called *Service-Level-Agreement (SLA)* [14]. In the simplest case, an SLA is defined in an n-dimensional space where each dimension represents a *service performance metrics*. For each such metrics a *service level objective* is defined (what an agent is expected to achieve) – e.g. by means of a threshold vector with n dimensions. The actual performance of an agent is evaluated in respect to these objectives. An experience can for instance be regarded as positive, if the

actual performance lies above the objectives in all dimensions, and as negative otherwise.

An SLA has the advantage that it formally specifies what is "good" and "bad" behavior. However, it has several limitations in the context of trust. First, it does not handle the uncertainty that is inherent in observations. Secondly, it does not account for side-effects or proactive behavior unless explicitly specified (see below). Finally, an SLA does not ask for the reasons for a divergence between the objectives and the actual performance. In particular, it does not account for the mental state of the performing agent as Castelfranchi and Falcone [2] claim. The distinction between incompetence of an agent and malicious behavior is not possible. We therefore stress the importance of a *sophisticated* component for making experiences, that amongst others comprises cognitive modeling. It could be argued that errors in this step of interpretation can be summed out statistically. This is not correct because already few errors together with the self amplifying property of trust [15] can completely change the evolution of a trust-based system.

In conclusion, the process of interpretation should, on the one hand, accurately reflect the trustor's satisfaction with the outcome of an interaction. On the other hand, an interpretation should evaluate the information with respect to how the assessed agent will behave in future. A component for interpretation that incorporates both facets faces some challenges. In the following we examine these challenges one by one and refer to literature where appropriate.

Intention recognition. One challenge for the step of interpretation is to distinguish between *malicious*, *selfish* and *incompetent* behavior. This distinction can be essential for reasoning about the future behavior of an agent. Malicious behavior for instance can cause a stronger decrease in the trust towards the agent – independent of the *context* (for details on context see Sect. 2.3)!. For being able to make a distinction, the intention of an agent has to be determined. An accurate intention recognition is amongst others important to address the first time offender problem (see also Sect. 2.3).

Mao and Gratch [16] and Demolombe and Fernandez [17] presented two approaches to infer another agent's goals and plans based on observations (the latter assume complete knowledge of the actions performed). The authors call this process *intention recognition*.

Excusableness of failures. One reason for interpreting a failure of an agent falsely as negative experience is *excusableness*. Staab and Engel [18] proposed to call a failure *excusable* in respect to a *reference set* of agents, and within a time-frame $[t_1, t_2]$, iff all of the following three conditions are fulfilled:

1. The environment interfered with the performance of the task.
2. All other agents from the *reference set* would have failed to perform the task under the same conditions.
3. The performing agent tried its "best" within $[t_1, t_2]$ to accomplish the task.

The social facet of this problem is covered by the "social credit assigning problem", analyzed by Mao and Gratch [19]. They presented a computational

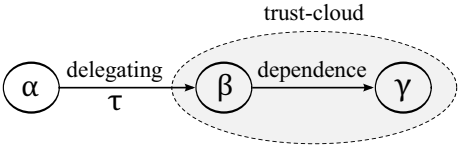

Fig. 2. Agents β and γ form a *trust-cloud*

approach to find out which agent is socially responsible to what extent for an action that he had executed. They follow Weiner [20] and use (1) the hierarchical structure of *coercion* to identify the responsible agents and (2) *intention* and *foreseeability* to determine the "intensity" of the responsibility.

Trust-clouds. Assume, an agent α delegates a task τ to an agent β; however, agent β depends in its execution of τ on a third agent γ (e.g. in order to accomplish τ, β needs units that only γ can construct). We say, that β and γ form in this case a trust-cloud towards α (see also Figure 2). A trust-cloud is not restricted to contain two agents, because γ again could depend on a further agent (actually, a whole graph of dependencies can occur here). Trust-clouds are for instance characteristical for *supply chain* settings [21]. There, customers purchase items from suppliers, which by themselves may be customers of other suppliers.

Now assume that in the trust-cloud shown in Figure 2, γ fails (and the failure is not excusable). It is not obvious how the trustworthiness of both agents β and γ towards α are involved in this process. First, it can be noted that if α does not know about the dependency between β and γ, he will completely blame β for the failure and will not be able to see that actually γ failed. If α does know about the dependency, the interpretation of the situation depends on whether there would have been alternatives for β than to rely on γ. Also, β himself did not fail in performing τ, so α should not trust less in β concerning β's abilities to perform τ. But in future, β might still depend on γ, and therefore, α should consider alternatives to β even if it trusts much in β's abilities. Therefore, we argue for the notion of "trusting in a trust-cloud" in settings like above – instead of trusting in an individual agent.

Side-effects. *Side-effects* that are caused by the performance of an agent, can influence its trustworthiness. Assume for example, an agent β successfully accomplishes a task τ in favor of agent α. However, agent β's actions produce side-effects that have a negative impact on agent α. Then, α should trust β more in how close he sticks to a deal (positive experience). At the same time, α should be less content about β's way to perform task τ (negative experience). This shows that one interaction can produce several experiences at the same time.

Note that side-effects can appear heavily delayed which can raise two problems. First, over time it gets harder and harder to establish a relationship between the performance of agent β and the side-effects his actions have produced.

Second, the side-effects are possibly not yet observed when trust reasoning is done (e.g. in the case of *global warming*).

Proactive behavior. As for side-effects, an interpretation should account for *proactive behavior*. Assume agent α asks agent β to accomplish task τ_1; but β decides to *proactively* accomplish task $\tau_2 \neq \tau_1$ as he believes, that in this case the resulting state is more desirable for α than in the case of τ_1. Assume further, agent β succeeds and his belief turns out to be true. Agent α can be unhappy about the fact that β did not exactly do what he asked for; in this sense it is a negative experience and the respective trust is diminished. But he is happy about β's intelligence and it is a positive experience in this sense. As in the case of side-effects, α makes several experiences in one interaction here.

Incomplete information. Monitored data can be incomplete or distorted and therefore an experience can comprise a certain amount of *uncertainty*. If possible, mechanisms should be provided that help to reduce this uncertainty. For example, whenever only the outcome of a performance but not the performance itself can be perceived, an agent can reason about the causes for the outcome. The huge area of research concerned with this problem is called causality (e.g. see Pearl [22]). To show its practical relevance in the context of trust, we give a concrete example in the area of networking. Assume agent α asks β to upload a signed data file to a server. As it is problematic to observe β's actions, α can try to download the file using another route; if this is possible and the file can be verified, β must have succeeded. If it is impossible to reduce the uncertainty of interpretation, it is propagated to the step of computation which is described in the following.

Unverifiable information. In certain scenarios, it cannot be decided whether the performance of another agent should be interpreted as positive or negative experience. This is mainly the case when the performance of an agent consists in the provision of information that cannot be tested for correctness by the receiver agent. A typical example are grid settings, where agents outsource computations to other agents in order to save resources. The received results for these outsourced computations cannot be verified as this verification would be as costly as the computation itself (see [23]). In [24], the authors propose therefore to challenge other agents with computations for which the correct result is already known. Based on the performance on these challenges, the correctness of the information can be estimated, and the trustworthiness can indirectly be assessed.

2.3 Step 3 – Computation

Given a set of experiences, a trust update algorithm tries to extrapolate the data into the future. The algorithm outputs some measure, usually a trust-value – a number in a continuous or discrete interval –, that represents a positive or negative forecast for future interactions with the trustee. For prediction, for instance

time series methods (e.g. *trend estimation*), *econometric methods* (e.g. *regression analysis*) or *probabilistic forecasting* can be used. But also game-theoretic reasoning can help to determine how an agent will behave, or what strategy he will follow next. Subjectivity is present also in this step: Parameters of the model can determine whether the agent is optimistic or pessimistic, how unforgiving he is (i.e. how heavy old experiences should be weighted with a so called *aging factor*), etc. In the following, we concisely survey the challenges that are specific to these algorithms.

First time offender problem. Think of an agent who pretends trustworthiness in many interactions, and as soon as other agents have developed a strong trust in him, he behaves maliciously. This problem is also called the *first time offender problem* [25]. In one extreme case, experiences accurately reflect intentions. Then, the first time offender problem does not exist. In the other extreme case, experiences ignore intentions. Then the first time offender problem cannot be tackled by any trust update algorithm. So, one key of coping with the first time offender problem lies in an accurate recognition of an agent's intention (interpretation, see Sect. 2.2). The other key is in a correct prediction of how the intention of the agent will evolve (computation).

Changes in a character. The first time offender problem just discussed is about a change from a feigned character to the real character. Opposed to that, the real character of an agent can change, too. This can be due to a change of the agent's strategy, his goals, his beliefs, etc. If a character changes gradually, trust models can for instance use linear prediction techniques to predict this change. If a character changes suddenly, it will take some time until this change is recognized. The reason for that is that a sudden change in a character can for a certain time not be distinguished from statistical outliers.

Change of identity. In open systems, agents can freely leave the system and reenter with a new identity. This enables agents, once they are not much trusted anymore by other agents, to get rid of their past and get a clean record again, simply by "logging of" and "logging in" with a new identity.

There are mainly two approaches to this problem. The first approach is to assign the lowest possible trust value to agents that (re-)enter a system. This way, agents that have a "clean record" have also a record that is not better than the record of any other agent. This forces agents to invest a lot to get trusted again. And, in the best case, the resulting costs make it unprofitable for malicious agents to change their identity.

The second approach is called *identity recognition* [25]. The identity of an agent is recognized on the basis of certain characteristics which are assumed to be the same after leaving and reentering a system. An example for such a characteristic would be the sending characteristics of a wireless network card (see [26]).

Undecidability. Trust reasoning that is based on experiences has one fundamental limit. This limit becomes evident in the context of *confidentiality* which

is illustrated by the following example. Assume, agent α transfers confidential information to agent β and both reach a *non-disclosure agreement (NDA)*, i.e. a contract that prohibits β to transfer the information to a third party. In the case that α detects a violation of the agreement by β, the respective indications or observations can be interpreted as negative experience. In the case that α does not detect anything, α also makes no experiences and cannot decide about β's trustworthiness, because α does not know whether he maybe simply missed outgoing messages of β. The same problem exists also whenever tasks are delegated without time limit. At no point in time it is known whether the agent will or will not accomplish the task (only in case of accomplishment). Basically, we face the *halting problem* [27] or the problem of verifying a scientific theory.

Context-sensitivity. Most researchers agree on the fact that trust is *context-sensitive*. This means that experiences in a certain context are only significant for trust-based decisions in similar contexts. Therefore, trust values should always be associated to a specific context. In literature, the term "context" is used in two different ways. Sometimes "context" refers to the type of task that is to be performed [28,29]. McKnight and Chervany [30] give the example that "one would trust one's doctor to diagnose and treat one's illness, but would generally not trust the doctor to fly one on a commercial airplane". And sometimes the term "context" refers to the conditions under which a task is performed [25]. The first notion refers only to the internal causes of an agent's performance [15] whereas the second notion refers to an interplay of internal and external causes. It is obvious that the two meanings are relevant for trust and so a trust-model should account for both of them – even though in different ways.

Incomplete information. Although some experiences with full certainty can be given, a prediction on the basis of these certain experiences will always remain uncertain. This is especially the case when the conflict within the experiences is high [4]. Therefore a distinction of the certainty of *observations/experiences* and *trust values* is necessary.

2.4 Step 4 – Decision-Making

The process of Decision-Making is the most complex process of the four processes described in this work. As a result of Decision-Making an agent knows how he is going to act. For example, an agent has to decide whether he is going to execute some task on its own or to delegate the task to another agent. Trust can be an important criterion for such a decision. If for instance no trusted agent is available, the agent may decide to execute a sensitive task on its own.

In the following, we will concentrate on two main problems of Decision-Making that arise especially when trust is used as a criterion in the decision process.

Trust dynamics. The behavior of an agent affects how other agents behave towards him. Even if one is not sure that a certain agent is trustworthy, it might pay off to delegate a task to this agent in order to show him that he is thought to be trustworthy. The hope of a confidence-building measure is that the trustee will

reciprocate the trust in future interactions. Such a behavior aims at exploiting the self-amplifying nature of trust (*trust begets trust* [31]). Analogously, a non-delegation of a task can lead to a decrease in the assessments of the trustor's trustworthiness and should therefore be well thought out.

Exploitation vs. exploration. Assume a system with 100 agents. Agent α knows 20 of the agents very well, 30 of them a bit, and 50 of them not at all. Of the 20 agents he knows very well, 50% are quite trustworthy, the other 50% are not (the same ratio is given for the 30 less known agents). In case α wants to delegate a task, he has to choose an agent from these 100 agents (let's ignore the context-sensitivity of trust for now). Is it better for α to choose one of the 10 well known and rather trustworthy agents, to choose one of the less known but possibly more trustworthy agents, or to choose one of the agents who are not known at all but who might be more trustworthy than all other agents? In other words, should an agent exploit the trust we have in known agents or should we explore unknown agents in order to find agents that are more trustworthy than those we know? These questions suggest that a good balance between exploitation and exploration is desirable. How a good balance for the *exploitation vs. exploration* dilemma can be found, is addressed in literature [32,33,34]. Alternatively, Reches et al. [35] propose to collect "the right amount" of witness-information about a set of alternative agents before delegating a task to one of them.

3 Integration of Witness-Information

In this section we discuss how the exchange of witness-information fits in our concept. Witness-information is any information which is exchanged between two trustors and which concerns the trustworthiness of a trustee. This information helps trustors to overcome the stage where not enough direct experiences with a certain trustee could be made yet, and also lets agents base their "trust"-reasoning on a larger knowledge base. This witness-information can also be used for computing the reputation a certain individual has among a set of agents.

As Figure 3 reveals, our concept allows two agents α and β to exchange witness information on all three "layers" of the trust reasoning process. It holds

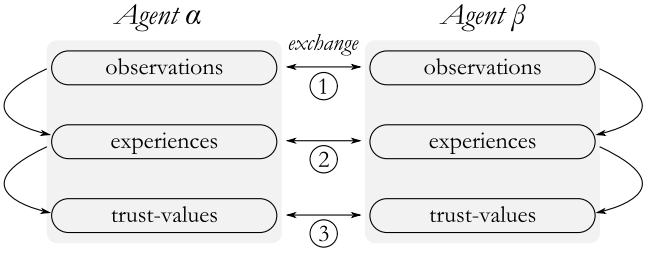

Fig. 3. Exchange of witness-information

that the higher the "layer" for information exchange is (that is the more often the information has been transformed), the higher is the level of subjectivity. This is the reason why in [13] witness-information is exchanged on the lowest possible layer, the layer of observations (1). This way each agent can make its own interpretation of observations that actually another agent made. To make this possible, observations have always to be bundled with context-information that allows for an interpretation.

Given that two agents believe that they have a similar way to monitor the world (e.g. they use similar sensors), and they trust each other, then they can exchange observations they have made (1). If two agents want to exchange witness-information on the layer of experiences (2), they have to make sure that they interpret observations in a similar way – or use a mechanism that transforms the subjective experience of one agent into the experience as the other agent would have interpreted it. Similarly, if two agents exchange witness-information on the layer of trust-values (3), they additionally need a similar subjectivity in the step of interpreting experiences – or again a mechanism for transformation. In general, it is unsafe to use information exchanged on the higher layers without knowing about the witness's subjectivity. Therefore, Abdul-Rahman and Hailes [36] introduced the concept of "semantic distance" between two agents that exchange information on the layer of trust-values: Only, if the agents feel that they have a similar way of looking at others, they exchange trust information.

In order to use witness-information on any layer, mechanisms are required that detect inaccurate information sources, i.e. agents that provide inaccurate information either because they are incompetent or they lie. Numerous approaches to this problem can be found in literature [7,11,37].

Also, the problem of *correlated evidence* has to be addressed when exchanging witness-information [38]. This means, that if different agents independently provide information about the same event, then the agent who uses this information has to merge the information and handle it as one experience instead of ten different experiences (about ten different events).

This work is not about reputation, however we want to note that, in contrast to trust, reputation unifies the subjectivity of a set of agents. In other words, the reputation an agent is assigned to by a set of agents, reflects the subjective view of each of these agents. As described above, for trust this subjectivity has to be filtered out from witness-information.

4 Conclusion and Future Work

We presented a modular concept for experience-based trust reasoning which integrates the cognitive and computational approaches. As a result of analyzing each component of the concept in more detail, we gave an extensive survey on challenges and limits that are specific to this kind of trust reasoning. Finally, we showed how witness-information can be integrated in our approach. We argue for exchanging different kinds of witness-information at the same time in order to improve the accuracy of the trust reasoning process.

In order to utilize the proposed concept as modular framework, interfaces have to be defined that specify the transfer of information from one component to another. A formalization of these interfaces will be part of future work. For this formalization it is important to account – throughout the whole process – for context-information and uncertainty.

References

1. Castelfranchi, C., Falcone, R.: Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In: ICMAS 1998: Proc. of the 3rd Int. Conference on Multi-Agent Systems, pp. 72–79. IEEE Computer Society, Los Alamitos (1998)
2. Castelfranchi, C., Falcone, R.: Trust is much more than subjective probability: Mental components and sources of trust. In: HICSS 2000: Proc. of the 33rd Hawaii Int. Conf. on System Sciences. IEEE Computer Society, Los Alamitos (2000)
3. Falcone, R., Castelfranchi, C.: Social trust: a cognitive approach. Trust and deception in virtual societies, 55–90 (2001)
4. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: IJCAI 2007: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence, pp. 1551–1556 (2007)
5. Capra, L., Musolesi, M.: Autonomic trust prediction for pervasive systems. In: AINA 2006: Proc. of the 20th Int. Conf. on Advanced Information Networking and Applications, vol. 2, pp. 481–488. IEEE Computer Society, Los Alamitos (2006)
6. Esfandiari, B., Chandrasekharan, S.: On how agents make friends: Mechanisms for trust acquisition. In: Proc. of the 4th Workshop on Deception, Fraud and Trust in Agent Societies (at AAMAS 2001), pp. 27–34 (2001)
7. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Auton. Agents Multi-Agent Syst. 12(2), 183–198 (2006)
8. Witkowski, M., Pitt, J.: Objective trust-based agents: Trust and trustworthiness in a Multi-Agent trading society. In: ICMAS 2000: Proc. of the 4th Int. Conf. on Multi-Agent Systems, pp. 463–464. IEEE Computer Society, Los Alamitos (2000)
9. Zacharia, G.: Collaborative reputation mechanisms for online communities. Master's thesis, Massachusetts Institute of Technology (1999)
10. Sabater, J., Sierra, C.: REGRET: Reputation in gregarious societies. In: Proc. of the 4th Workshop on Deception, Fraud and Trust in Agent Societies, pp. 194–195. ACM Press, New York (2001)
11. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. Auton. Agents Multi-Agent Syst. 13(2), 119–154 (2006)
12. Haas, Z.J., Deng, J., Liang, B., Papadimitratos, P., Sajama, S.: Wireless ad hoc networks. In: Perkins, C.E. (ed.) Ad Hoc Networking, pp. 221–225. Addison-Wesley, Reading (2001)
13. Şensoy, M., Yolum, P.: Ontology-based service representation and selection. IEEE Trans. Knowl. Data Eng. 19(8), 1102–1115 (2007)
14. Marilly, E., Martinot, O., Betgé-Brezetz, S., Delègue, G.: Requirements for service level agreement management. In: IPOM 2002: Proc. of the IEEE Workshop on IP Operations and Management, pp. 57–62 (2002)

15. Falcone, R., Castelfranchi, C.: Trust dynamics: How trust is influenced by direct experiences and by trust itself. In: AAMAS 2004: Proc. of the 3rd Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, pp. 740–747. IEEE Computer Society, Los Alamitos (2004)
16. Mao, W., Gratch, J.: A utility-based approach to intention recognition. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) AAMAS 2004. LNCS, vol. 3394. Springer, Heidelberg (2005)
17. Demolombe, R., Fernandez, A.M.O.: Intention recognition in the situation calculus and probability theory frameworks. In: Toni, F., Torroni, P. (eds.) CLIMA 2005. LNCS, vol. 3900, pp. 358–372. Springer, Heidelberg (2006)
18. Staab, E., Engel, T.: Formalizing excusableness of failures in multi-agent systems. In: PRIMA 2007: Proc. of the 10th Pacific Rim Int. Workshop on Multi-Agents, pp. 124–135 (2007)
19. Mao, W., Gratch, J.: The social credit assignment problem. In: Rist, T., Aylett, R.S., Ballin, D., Rickel, J. (eds.) IVA 2003. LNCS, vol. 2792, pp. 39–47. Springer, Heidelberg (2003)
20. Weiner, B.: The Judgment of Responsibility. Guilford Press, New York (1995)
21. Lambert, D.M., Cooper, M.C.: Issues in supply chain management. Industrial Marketing Management 29, 65–83 (2000)
22. Pearl, J.: Causality: models, reasoning, and inference. Cambridge University Press, Cambridge (2000)
23. Staab, E., Fusenig, V., Engel, T.: Using correlation for collusion detection in grid settings. Technical Report 000657499, University of Luxembourg (July 2008)
24. Staab, E., Fusenig, V., Engel, T.: Towards trust-based acquisition of unverifiable information. In: Klusch, M., Pěchouček, M., Polleres, A. (eds.) CIA 2008. LNCS, vol. 5180, pp. 41–54. Springer, Heidelberg (2008)
25. Rehák, M., Pechoucek, M.: Trust modeling with context representation and generalized identities. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS, vol. 4676, pp. 298–312. Springer, Heidelberg (2007)
26. Stefano, A.D., Terrazzino, G., Scalia, L., Tinnirello, I., Bianchi, G., Giaconia, C.: An experimental testbed and methodology for characterizing IEEE 802.11 network cards. In: WoWMoM 2006: Proc. of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 513–518. IEEE Computer Society, Los Alamitos (2006)
27. Sipser, M.: Introduction to the Theory of Computation. Int. Thomson Publishing (1996)
28. Grandison, T., Sloman, M.: Trust management tools for internet applications. In: iTrust 2003: Proc. of the 1st Int. Conf. on Trust Management, pp. 91–107. Springer, Heidelberg (2003)
29. Kinateder, M., Baschny, E., Rothermel, K.: Towards a generic trust model - comparison of various trust update algorithms. In: iTrust 2005: Proc. of the 3rd Int. Conf. on Trust Management, pp. 177–192. Springer, Heidelberg (2005)
30. McKnight, D.H., Chervany, N.L.: The meanings of trust. Technical report, University of Minnesota (1996)
31. Bradach, J.L., Eccles, R.G.: Price, authority, and trust: From ideal types to plural forms. Annu. Rev. Sociol. 15, 97–118 (1989)
32. Dearden, R., Friedman, N., Andre, D.: Model based bayesian exploration. In: UAI 1999: Proc. of the 15th Conf. on Uncertainty in Artificial Intelligence, pp. 150–159 (1999)

33. Chalkiadakis, G., Boutilier, C.: Coordination in multiagent reinforcement learning: a bayesian approach. In: AAMAS 2003: Proc. of the 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems, pp. 709–716. ACM, New York (2003)
34. Teacy, W.T.L., Chalkiadakis, G., Rogers, A., Jennings, N.R.: Sequential decision making with untrustworthy service providers. In: AAMAS 2008: Proc. of the 7th Int. Conf. on Autonomous Agents and Multiagent Systems, pp. 755–762 (2008)
35. Reches, S., Hendrix, P., Kraus, S., Grosz, B.J.: Efficiently determining the appropriate mix of personal interaction and reputation information in partner choice. In: AAMAS 2008: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems, pp. 583–590 (2008)
36. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: HICSS 2000: Proc. of the 33rd Hawaii Int. Conf. on System Sciences. IEEE Computer Society, Los Alamitos (2000)
37. Whitby, A., Jøsang, A., Indulska, J.: Filtering out unfair ratings in bayesian reputation systems. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) AAMAS 2004. LNCS, vol. 3394. Springer, Heidelberg (2005)
38. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: AAMAS 2002: Proc. of the 1st Int. Joint Conf. on Autonomous agents and multiagent systems, pp. 475–482. ACM, New York (2002)