A Social Trust Model for Services

Return to Published Papers

Manling Zhu
Academy of Mathematics and Systems Science,CAS, Beijing, 100080, P.R.China
zhumanling@amss.ac.cn

Lin Liu
School of Software, Tsinghua University, Beijing, 100084, P.R. China
linliu@tsinghua.edu.cn

Zhi Jin
AMSS, CAS, ICT, CAS, Beijing, 100080, China
zhijin@math.ac.cn

Abstract

This paper proposes a formalism for the Trust requirements modeling framework, which can be used as a means of studying the trustworthiness of service-oriented environments. We argue that a modeling framework, representing explicitly the underlying assumptions, essential factors, and reasoning rules about Trust, can not only describe the problem domain but also be reasoned formally. This model offers better understanding to the Trust relationships in a web services world. Eventually, it will assist participants of an open service network in making rational communication and operation decisions.

1. Introduction

As the open service-oriented information systems are widely used, trust is becoming a central issue in this increasingly networked environments. Techniques for system analysis and design have, in the past, been focused primarily on addressing functional requirements, assuming that all parties are trusted [12]. However, given today's environments, it is inevitable to present new techniques to bring issues of non-functional requirements, such as trust, into the system analysis and design process. For example, in electronic commerce, exchanges often take place among parties unfamiliar to each other because of the openness of the environment. That is, a participant can join and leave this network at any given time and he could do perform any action. For example, a party could sell products, leave the system as soon as he gets paid, or he could also buy other products from another party in the same system after selling, and then leave to a different system. Moreover, an open service network allows participants having their own aims and objectives with different characteristics (e.g. policies, abilities, roles) to enter the system and interact with one another. It is convenient to take the party described above as an example. That party could then join the second auction system as a new bidder although he sold goods in the first system before. Given this, participating parties are likely to be faced with a number of possible interaction partners with varying properties. In such complex environment deception and fraud are possible because participants would encounter lots of actors which might have conflicting interests or malicious intent. Since the parties communicating frequently in multiple ways often do not have enough knowledge on each other before the actual interaction, it is ineluctable for participants to measure the trustworthiness in order to make relatively appropriate operation decisions.

One of the typical open service environments is the market place on the Web as follows.

In a market place on the Web, there are lots of suppliers and purchasers. The suppliers on the Internet are intend to sell goods, while the purchasers intend to buy the expected staff online to fulfill their requirements. All the participants need to interact with each other to serve their own purposes, and they have to consider the trustworthiness of the co-operators. For the suppliers, they want to estimate whether the purchasers are trusted in buying the commodities from them, similarly for the purchasers, they need to measure whether the suppliers are reliable to provide the expected goods for them.

We require a trust model to describe the scenario above in which trust is evolved among the participating parties. It obviously could raise the problems such as: how much trust do the purchasers need while they buy staff online? How much trust should the suppliers have when they provide goods to the exchange partner? Should the sellers trust that the buyer is able to pay for goods or services? Which actors to choose for the participants could maximize their own interests from the exchanging? All these are just a few questions related to trust in the market place on the Web. Therefore, it is inevitable to measure the trustworthiness in order to make relatively appropriate operation decisions. In the open network, the traditional conceptual models may not be very helpful.

A number of efforts exist in modeling trust in the open environment. These trust models can be classified as either learning (and evolution) based, reputation based, or socio-cognitive based [6]. While the learning and evolutionary models aim to endow parties with strategies that can cope with lying and non-reciprocative participants [13], reputation models enable parties to gather information in richer forms from their environment and make rational inferences from the information obtained about their counterparts [14]. Socio-cognitive models adopt a rather higher level view of trust that takes the knowledge of motivations of other parties for granted and proposes ways to reason about these motivations [1], [2], [3], [4]. However, mathematical formalisms offer analysis, but these approaches require strong assumptions, and are only good for specialized, idealized environments, while practical approaches have no analysis and hard to adapt. As a result of these deficiencies, what is needed in the open Internet environments is a formalism for the Trust requirements modeling which can capture dynamic trust relations among participants. It should support formal reasoning and should have the ability to deal with interactions between technology and human social behavior.

In this paper, we present a modeling framework, representing explicitly the underlying assumptions, essential factors, and reasoning rules about social trust, which can not only describe the problem domain but also conduct formal reasoning and measurement to trustworthiness in a distributed web service environment. It aims to assist participants of an open network in making rational operation decision. The rest of the paper is organized as follows. Section 2 gives precise definitions as basic terms in our social trust requirements modeling framework, followed by the rules for formal reasoning. Section 3 analyzes our model in two situations with typical social rules. Section 4 discusses a case of a market place on the Web demonstrating the trust modeling framework we proposed. Section 5 concludes the paper.

2. Service trust model

We consider for example the definition of trust provided in the article of Castelfranchi: *Trust is much more than subjective probability* [4]. In his social trust of view, the definition of *Trust is subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends* (translation from Italian) [13] is correct, and it stresses that trust is basically an estimation, an opinion, an evaluation [3]. However, it is not complete in some sense, since it just refers to one dimension of trust (predictability), while ignoring the other dimensions, such as competence, requirement, persistence etc., and it does not explain what is a trust evaluation made of and based on: in fact, the subjective probability mashes together too many important parameters and trust factors, which are very relevant in social reasoning. Considering that web services are software agents which have their own requirements to fulfill, and enough knowledge on how to achieve their goals, we believe that Trust in the service networks is a belief and as a social attitude and relation. Actually, in this social world, the interaction and co-operation between participating agents indicate their social nature, including cognitive attributes and human belief. In this paper, we aim towards building a generic formal ontology to represent trust relations between participants, which based on the social trust, and analyze what the essential factors forming the trust relationships between agents are, what relations among these factors are, what constrains need to be satisfied, and how to integrate the factors for studying the trustworthiness of the co-operated agents.

This section is divided into two parts. In the first part, we will present the conceptual structure of social trust based service trust model. In the next part some reasoning rules will be proposed based on the conceptual structure.

2.1. Conceptual structure

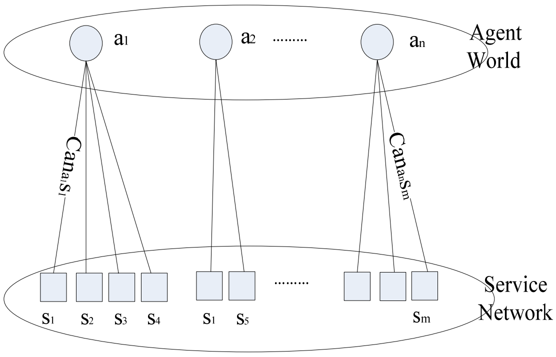

Figure 1. The structure of Service network

A basic concept related to trust is the agent who serves as the trustor or the trustee having his own intentions, cognitive abilities and knowledge on how to fulfill his requirements.

We give the service network structure in Figue1.

Definition 1. An agent is an active entity that carries out actions to fulfill his requirements by updating his knowledge.

We usually represent it as: ***Agent(a)***, where *a* is an agent.

All the agents in an open environment form an agent society, which is called a set of agents.

We usually use *AGENT* to denote the agent society, i.e $AGENT=\{a_1, \ldots, a_n \mid \boldsymbol{Agent(a_i)}, i\in[1,n]\}$.

In a market place on the Web, suppliers and purchasers are both agents according to definition 1, and all the participants in the market place on the Web form an agent society.

In the trust relationships, the reason why a participant trusts the other one is relying on the trustworthiness of the service provided by the trustee. Once a service can not fulfill the requirements of the requestor, from his point of view, the trustworthiness of the service provider must decrease.

Definition 2. A service is used to represent the functions performed by agents.

We usually represent it as: ***Service (s)***, where *s* is a service.

All the services form a service network, which is called a set of services.

We usually use *SERVICE* to denote the service network, i.e $SERVICE=\{s_1, \ldots, s_m \mid \boldsymbol{Service(s_i)}, i\in[1,m]\}$.

For instance, selling goods to purchasers is a service provided by suppliers, while buying commodities from suppliers is a service of purchasers in the market place on the Web scenario. All the dealing actions performed by the agents including buying and selling form a set of services.

If the requestor believes that the requested agent is capable of providing the expected service, the trustworthiness of this service provider increases.

Definition 3. If agent *a* has the capability of providing the functions of service *s*, then *a* can do service *s*, which is represented as: $\boldsymbol{Can}_a\, s$.

For each agent $a\in A$, let $CANa = \{s_1, \ldots, s_l \mid \boldsymbol{Can}_a\, s_i, i\in[1,l]\}$ representing the set of Capable services of *a*.

As shown in Figue1, the set of Capable services of a_1 is $CANa_1 = \{s_1, s_2, s_3, s_4 \mid \boldsymbol{Can}_{a1} s_i, i\in[1,4]\}$, Similarly, the set of Capable services of a_2 is $CANa_2 = \{s_1, s_5 \mid \boldsymbol{Can}_{a2}\, s_1, \boldsymbol{Can}_{a2}\, s_5\}$.

In a market place on the Web, suppliers may have the capability of selling the expected goods or not, while purchasers have the right to or not to buy commodities they need.

Though an agent can provide a service, it is unsure that this agent will actually perform it.

Definition 4. If agent *a* carries out the functions of service *s*, then *a* actually performs service *s*, which is represented as: $\boldsymbol{Perform}_a s$.

For each agent $a\in A$, let $PERFORMa= \{s_1, \ldots, s_p \mid \boldsymbol{Perform}_a s_i, i\in[1,p]\}$ representing the set of Performing services of agent *a*.

For example, in a market place on the Web, the suppliers may not be willing to provide the goods that the purchasers expect. It means that they will not perform the desired service to the purchasers.

Whether a service will be actually performed is not only depending on the capability of the providing agent, but also related to the motivation on doing the expected service of that requested one. If the requirements of the providing agent can also be fulfilled by doing this service, the probability of performing it will increase. On the contrary, if an agent is capable of doing the service but which can not satisfy its requirements, this agent may not provide the expected service.

Definition 5. If agent *a* needs the functionality of service *s*, then agent *a* requires service *s*, which is represented as: $\boldsymbol{Require}_a\, s$.

For each agent $a\in A$, let $REQUIREa=\{s_1, \ldots, s_q \mid \boldsymbol{Require}_a s_i, i\in[1,q]\}$ representing the set of Required services that agent *a* needs.

In a market place on the Web, suppliers and purchasers both require the business to be performed for their own purposes. The suppliers want to sell goods for gaining profit, and the purchases need the commodities they expected.

Some agents promise to provide an expected service after the service request, but it is also not sure that they will keep their words to perform the service.

Definition 6. If agent *a* promises to provide service s, then agent *a* commits to perform service s, which is represented as: $\boldsymbol{Commit}_a\, s$.

For each agent $a\in A$, let $COMMITa=\{s_1, \ldots, s_t \mid \boldsymbol{Commit}_a s_i, i\in[1,t]\}$ representing the set of services that agent *a* Commits to provide.

For example, the suppliers on the Internet need the purchasers who they will trade with to commit to buy the goods from them, but the purchases also require the suppliers to commit to sell the expected goods.

After representing the factors above, we turn to define the trust relationship. As trust is an attitude from a service requestor to the provider about the particular services, the representation of trust include trustor, trustee, and service three parameters.

Definition 7. If agent *a* trust agent *b* in providing service *s*, then there is a $\textbf{\textit{Trust}}_a\,(b,s)$. If agent *a* distrust agent *b* in providing service *s*, then there is a $\neg\textbf{\textit{Trust}}_a\,(b,s)$.

In particular, if agent *a* trusts himself in providing service *s*, then we can represent it as: $\textbf{\textit{Trust}}_a\,(a,s)$.

Definition 8. We use $TRUST=\{(a,b,s),\ a,b\in A,\ s\in S\}$ and $\textbf{\textit{Trust}}_a\,(b,s)\}$ to represent trust structure.

If $(a,b,s)\in TRUST$, there is a $\textbf{\textit{Trust}}_a\,(a,s)$. It means that agent *a* trusts agent *b* in providing service *s*.

In the case we study, suppliers may or not trust purchasers in buying commodities from them, while purchasers may trust suppliers in selling expected goods to them or not trust that.

2.2. Relationships among concepts

In this part we will give some trust reasoning rules based on the concept structure above.

In respect that we will model the trust relationship in the open environments, and our model is updated dynamically, we use operators of Temporal Logic into our reasoning rules, where ◇ means at least once, and □ means always.

An agent requesting a service should believe that the requested agent knows that it can do the service. That is, it is self-confident about providing the expected service. It is difficult to trust someone that does not trust himself [3]. The trust relations have two sides, but the self-confidence of an agent is a special case which emphasizes that an agent trusts in itself.

Rule 1. (Self-confidence Rule)

$$\textbf{\textit{Can}}_a\, s\wedge \textbf{\textit{Require}}_a\, s\Rightarrow \textbf{\textit{Perform}}_a\, s$$

That means when agent *a* is self-confident about providing service *s*, if agent *a* is capable of providing a service *s*, and it also has the needs of performing the service, it can perform this service.

If a providing agent who promises to perform a service is not stable enough in its intentions, it might change its mind due to its conflicting goals with the service or interfering from the outside, and then not actually provides the service. In contrast, if an agent is persistent to its intentions, once it promises to do a service, it will perform the expected service in any case.

Rule 2. (Persistence Rule)

$$\textbf{\textit{Can}}_a\, s\wedge \textbf{\textit{Commit}}_a\, s\Rightarrow \textbf{\textit{Perform}}_a\, s$$

(Non-persistence Rule)

$$\textbf{\textit{Can}}_a\, s\wedge \textbf{\textit{Commit}}_a\, s\Rightarrow \neg\textbf{\textit{Perform}}_a\, s$$

That first rule means when agent *a* is persistent, if agent *a* is capable of providing service *s*, and he also commits to do the service, then he will give the performance. That second rule means when agent *a* is not persistent, if agent *a* is also capable of providing service *s*, and he commits to do the service, but he may not give the performance.

It is important to analyze whether an agent is honest or not for estimating its trustworthiness. If an agent promises to the requestor on doing a service, but it has not the capability of this performance, that is, it tells a lie to the other agent. As a result of the lying action, its trustworthiness will decrease. In this way we elicit a set of two opposite rules.

Rule 3. (Honesty Rule)

$$\textbf{\textit{Commit}}_a\, s\Rightarrow \textbf{\textit{Can}}_a\, s$$

(Non-honesty Rule)

$$\neg\, \textbf{\textit{Can}}_a\, s\Rightarrow \textbf{\textit{Commit}}_a\, s$$

$$\neg\, \textbf{\textit{Can}}_a\, s\wedge \textbf{\textit{Commit}}_a\, s\Rightarrow \neg\, \textbf{\textit{Perform}}_a\, s$$

That rule means when agent *a* is honest, if it commits to provide a service, then it must be capable of doing this service. When agent *a* is dishonest, that is, it can not do service *s*, but it commits to provide *s*. What the result is that agent *a* can not perform service *s*.

Rule 4. (Motivation Rule)

$$\textbf{\textit{Commit}}_a\, s\vee \textbf{\textit{Require}}_a\, s\Rightarrow \textbf{\textit{Motivated}}_a\, s$$

In the rule above, $\textbf{\textit{Motivated}}_a\, s$ means that agent *a* has the motivation to provide service *s*. Rule 4 shows that if agent *a* commits to do or requires to do, then it will do.

According to the trust factors above, we can elicit the rules of trust evaluation. If a service provider was unreliable at one of the past interactions with the service requestor, its trustworthiness will decrease from the perspective of the requesting agent. If a service provider has been always reliable in the past experiences with the service requestor, its trustworthiness will increase in the requestor's opinion. Therefore, we use operators of Temporal Logic into our reasoning rules.

Rule 5. (Reliance Rule)

$$\textbf{\textit{Commmit}}_a^b\, s\wedge\Diamond\neg\textbf{\textit{Perform}}_a^b\, s\Rightarrow\neg\textbf{\textit{Reliable}}_b\,(a,s)$$

$$\textbf{\textit{Commmit}}_a^b\, s\wedge\Box\textbf{\textit{Perform}}_a^b\, s\Rightarrow\textbf{\textit{Reliable}}_b\,(a,s)$$

In these rules, $\textbf{\textit{Commmit}}_a^b\, s$ means that agent *a* commits to agent *b* of doing service *s*; $\Diamond\neg\textbf{\textit{Perform}}_a^b\, s$ means that agent *a* at least once did not perform the service *s* to the agent *b*; and $\Box\textbf{\textit{Perform}}_a^b\, s$ means that agent *a* always provides the service *s* to agent *b*. $\neg\textbf{\textit{Reliable}}_b\,(a,s)$ means that agent *a* considers that providing service *s* by agent *b* is unreliable. And the first rule means that agent *a* will not believe in the commitment made by agent *b*. ***Reliable**$_b$ (a,s)* means that agent *a* considers that providing service *s* by agent *b* is reliable. And the second rule means that agent *a* will believe in the commitment made by agent *b* more.

Rule 5 means if agent *a* once committed to agent *b* about providing service *s*, but he did not do the performance, then agent *b* considered that agent *a* was unreliable, in other words, agent *b* would distrust agent *a* about doing service *s*. If agent *a* always provides service *s* that he commits to agent *b*, then agent *b* considers that agent *a* is reliable, in other words, agent *b* will trust agent *a* about doing service *s*.

In accordance with Rule 5, we can give the reputation rule description about an agent.

Rule 6. (Reputation Rule)

$$\exists_{a,x\in A}, \neg\, \boldsymbol{Reliable}_x(a,s) \Rightarrow \downarrow \boldsymbol{Reputation}(a,s)$$
$$\exists_{a,x\in A}, \ \boldsymbol{Reliable}_x(a,s) \Rightarrow \ \uparrow \boldsymbol{Reputation}(a,s)$$

That rule means if one of agents in the environment except agent *a* considers that service *s* provided by agent *a* is unreliable, then there is a negative effect on the reputation of agent *a*. If one of agents in the environment except agent *a* considers that service *s* provided by agent *a* is reliable, then there is a positive effect on the reputation of agent *a*.

In the rule 6, $\downarrow$ ***Reputation**(a,s)* means that other agents will not believe agent *a* in doing service *s*, that is, the reputation of agent *a* falls; while $\uparrow$ ***Reputation**(a,s)* means that other agents believe agent *a* in doing service *s*, that is, the reputation of agent *a* rises.

We represent five dimensions of trust, which are self-confidence, persistence, honesty, motivation, reputation, and show the relations of trust factors in Figure2.

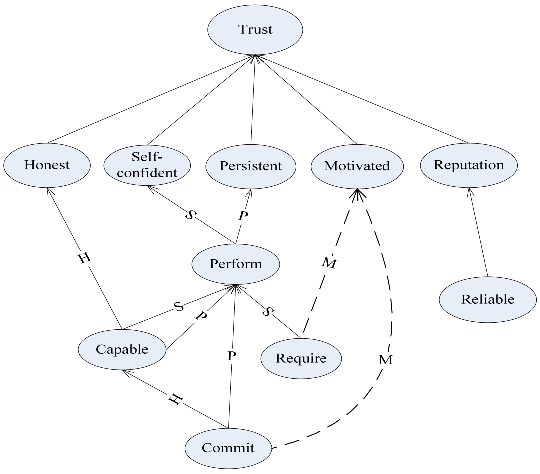

Figure 2. The structure of trust factors

According to the structure of trust factors and their relations, a positive increasing of one or more dimensions of trust will lead to a positive increasing of the trustworthiness. For example, if an agent is honest, then the trustworthiness of the agent will increase. Similarly, if an agent is persistent, then the trustworthiness of the agent will also increase. Other dimensions of trust affect the trustworthiness in the same way. By contraries, a negative increasing of one dimension of trust will lead to a negative increasing of the trustworthiness. For instance, if an agent has no motivation to do an expected service, then others must not trust that agent in co-operating with.

We give the trust updating process depending on the times of interactions in Figure3 which is the complement of the structure of trust factors above.

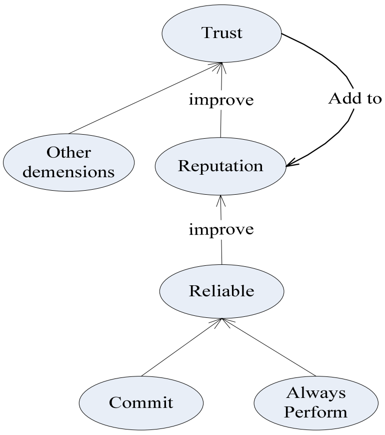

Figure 3. The updating process of trust

More detailed, we represent the evolving of the cognitive trust based on dynamic trust relationship between agents in Figure4.

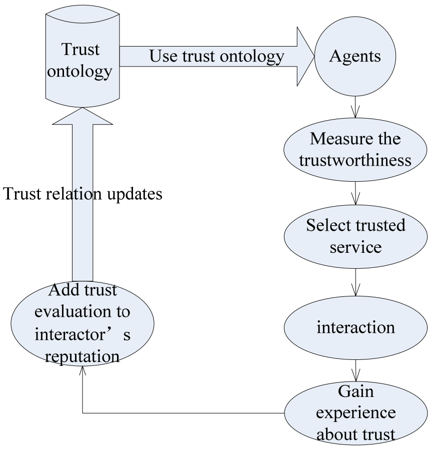
Trust ontology
Use trust ontology
Agents
Measure the trustworthiness
Trust relation updates
Select trusted service
Add trust evaluation to interactor’s reputation
interaction
Gain experience about trust

Figure 4. The evolving of trust evaluation

From the structure of trust factors and the evolving of trust evaluation above, we could see that the trustworthiness is updated dynamically. Participating agents use the trust ontology to choose the most trusted agent to interact depending on his trustworthiness. After one performance, the requestor adds the reliance evaluation of the co-operator to his reputation value so that an agent's reputation is updated. And then the agent does the next service selecting according to the evolved trust ontology with that updated reputation.

3. Trust analysis in two typical social rules

After given the trust model in the previous section, we will analyze it in two social rules' situations, which are the most familiar societies in current open networks. That is in order to assist in describing and reasoning the trustworthiness of the agents in these two service environments. As follows we apply the formalism to the networks with these typical social rules.

3.1. Completely selfless society

In this society, each of the agents is selfless so that everyone is willing to provide services regardless of any interests. It is no need to consider the cheat of any agent. Applying our trust model to this typical society is:

$\forall\ a \in A,$

$\textbf{\textit{Commit}}_a\, s \Rightarrow \textbf{\textit{Can}}_a\, s \Rightarrow \textbf{\textit{Perform}}_a\, s \Rightarrow \textbf{\textit{Reliable}}_a\, s$

$\Rightarrow \uparrow \textbf{\textit{Reputation}}(a, s) \Rightarrow \textbf{\textit{Trust}}\ (a,s)$

As we can see that it is no cheat of any agent in this selfless society. In other words, if an agent promises to do a service, then it must have the capability of providing the service, and it will actually do so. As a result, all agents are trusted in this environment.

3.2. Self-interested society

In this society, every agent is self-interested so that an agent will provide a service for another unless the requestor can do some other services to fulfill his requirements back. Trust relation is a rational interested connection in this environment. Applying our trust model to this typical society is:

$\forall\ a,b \in A,\ s_1,\ s_2 \in s,$

$\textbf{\textit{Can}}_a\, s_1 \wedge \textbf{\textit{Can}}_b\, s_2 \wedge \textbf{\textit{Require}}_a\, s_2 \wedge \textbf{\textit{Require}}_b\, s_1$

$\Rightarrow \textbf{\textit{Perform}}\,{}^{b}_{a}\, s_1 \wedge \textbf{\textit{Perform}}\,{}^{a}_{b}\, s_2$

$\Rightarrow \textbf{\textit{Reliable}}_b\,(a,s_1) \wedge \textbf{\textit{Reliable}}_a\,(b,s_2)$

$\Rightarrow\ \uparrow \textbf{\textit{Reputation}}\ (a,s_1) \wedge \uparrow\ \textbf{\textit{Reputation}}\ (b,s_2)$

$\Rightarrow \textbf{\textit{Trust}}_b\,(a,s_1) \wedge \textbf{\textit{Trust}}_a\,(b,s_2)$

where $\textbf{\textit{Perform}}\,{}^{b}_{a}\, s_1$ means that agent *a* performs service s_1 to agent *b*; $\textbf{\textit{Perform}}\,{}^{a}_{b}\, s_2$ means that agent *b* performs service s_2 to agent *a*.

As we can see the trust relationships from the trust reasoning above, that typical environment is self-interest oriented which is a relatively more rational society.

From the analyses we can see that the proposed formalism can be used to describe different domain assumptions, social rules in a service environment. On the basis of the two typical societies, in the next section we will discuss a generic world in which agents are more complicated.

4. Case study

To illustrate how the formalism we proposed can be used in an open service environment, we describe a generic scenario taken from the Web sale domain. There are lots of bargainors and purchasers on the Internet. Thus it is important to select the most reliable cooperator which is related to trustworthiness of him.

Suppose that suppliers *a* and *b* can sell commodity *M* on the Internet, while purchasers *p* and *q* need to buy *M*, then they order *M* from *a*. But *p* finds supplier *b* claiming that *b* can sell *M* in a lower price before *p* pays for *M* to *a*, so *p* gives up the trade-off with seller *a*, and changes to buy from seller *b*. Meanwhile, *q* sticks to his words and buys *M* from *a*. On the other hand, buyer p finds *b* dose not have any commodity M at the trade-off time. Finally *p* turns back to find seller *a*.

Applying our model to the scenario above, we can have the following reasoning process, where s_1 means buy *M* from *a*'s point of view, s_2 means sell *M* from *p*'s point of view.

1. From rule 2 we have:

$\textbf{\textit{Can}}_p\, s_1 \wedge \textbf{\textit{Commit}}_p\, s_1 \Rightarrow \neg\ \textbf{\textit{Perform}}_p\, s_1$

It shows that *p* is not persistent to his intentions, and he dose not perform s_1.

Then from rule 5 we have:

$\rightarrow \textbf{\textit{Commit}}\,{}^{a}_{p}\, s_1 \wedge \Diamond \neg \textbf{\textit{Perform}}\,{}^{a}_{p}\, s_1 \Rightarrow \neg \textbf{\textit{Reliable}}_a\,(p,s_1)$

It means that *a* considers that *p* is unreliable, in other words, *a* will distrust *p* in doing service s_1.

Then from rule 6 we have:

$\rightarrow \neg \textbf{\textit{Reliable}}_a\,(p,\ s_1) \Rightarrow \downarrow \textbf{\textit{Reputation}}\,(p,\ s_1)$

$\quad \Rightarrow \neg \textbf{\textit{Trust}}_a\,(p,\ s_1)$

It means that the reputation of *p* falls, that is, *p* will be distrusted of doing service s_1.

2. From rule 1 we have:

$\textbf{\textit{Can}}_q\, s_1 \wedge \textbf{\textit{Require}}_q\, s_1 \Rightarrow \textbf{\textit{Perform}}_q\, s_1$

It means that *q* is self-confident, then service s_1 provided by *q* is trusted.

Then from rule 2 we have:

$\mathbf{Can}_q\, s_1 \wedge \mathbf{Commit}_q\, s_1 \Rightarrow \mathbf{Perform}_q\, s_1$

It shows that *q* is persistent to his intentions, and his trustworthiness is increasing.

Then from rule 5 we have:

$\rightarrow \mathbf{Commit}_q^a\, s_1 \wedge \Box \mathbf{Perform}_q^a\, s_1 \Rightarrow \mathbf{Reliable}_a\,(q, s_1)$

It means that *a* considers that *q* is reliable, in other words, *a* will trust *q* in doing service s_1.

Then from rule 6 we have:

$\rightarrow \mathbf{Reliable}_a\,(q, s_1) \Rightarrow \uparrow \mathbf{Reputation}\,(q, s_1)$
$\quad \Rightarrow \mathbf{Trust}_a\,(q, s_1)$

It means that the reputation of *q* rises, that is, *q* will be trusted about doing service s_1.

3. From rule 3 we have:

$\neg\, \mathbf{Can}_b\, s_2 \Rightarrow \mathbf{Commit}_b\, s_2$

It means that *b* lies, and he is not honest.

From rule 3 we have:

$\rightarrow \neg\, \mathbf{Can}_b\, s_2 \wedge \mathbf{Commit}_b\, s_2 \Rightarrow \neg\, \mathbf{Perform}_b\, s_2$

It means that *b* does not perform s_2.

From rule 5 we have:

$\rightarrow \mathbf{Commit}_b^p\, s_2 \wedge \Diamond \neg \mathbf{Perform}_b^p\, s_2 \Rightarrow \neg \mathbf{Reliable}_p(b, s_2)$

It means that *p* considers that *b* is unreliable, in other words, *p* will distrust *b* in doing service s_2.

From rule 6 we have:

$\rightarrow \neg \mathbf{Reliable}_p\,(b, s_2) \Rightarrow \downarrow \mathbf{Reputation}\,(b, s_2)$
$\quad \Rightarrow \neg\, \mathbf{Trust}_p\,(b, s_2)$

It means that the reputation of *b* falls, that is, *b* will be distrusted about doing service s_2.

As the reasoning process shown, supplier *a* considers that purchaser *p* is unreliable, and will distrust *p* about doing service s_1. The reputation of *p* falls. Learning from this interaction, *a* will add the knowledge that *p* is distrusted into his belief. When *p* turns back to *a*, *a* will not co-operate with him. On the contrary, *a* will believe that *q* is trusted for the successful interaction experience before. In his point of view, *q* is more trusted than *p*, that is, *a* will select the former as his co-operator but avoids the latter in the next decision. Similarly, purchaser *p* considered that supplier *b* is unreliable, and will distrust *b* about doing service s_2. The reputation of *b* falls. From this interaction *p* can learn the knowledge that *b* is distrusted which will be added into his belief. In the next selection, *p* will not choose *b* to co-operate with. It is obvious that the evolved belief can assist the agent in making rational operation decisions.

The rules applied above are illustrating that the proposed formalism can be easily used and effectively analyze trust relations in an open environment.

5. Related works and conclusions

The approach proposed in this paper mainly inspired from Cristiano Castelfranchi. and Rino Falcone's social trust model [1], [2], [3], [4] which provides a definition of trust both as a mental state and as a social attitude and relation, and present the mental ingredients of trust: its specific beliefs and goals, with special attention to evaluations and expectations. In particular, they highlight the importance of a cognitive view of trust in contrast to a mere quantitative view of trust [6], which are learning based [13] and reputation based [14] approaches. As we mentioned above, learning based trust model can cope with lying and non-reciprocative participants, but its belief may only be acquired if the interaction is repeated a number of times such that there is an opportunity for the participants to learn their opponent's strategy or adapt to each other's strategy. While reputation models enable parties to gather information in richer forms from their environment and make rational inferences from the information obtained about their counterparts, but it does not handle the problem of lying (strategically) among parties. Ratings are obtained in a cooperative manner rather than in a competitive setting. In socio-cognitive models, the context they choose is that of task delegation where an agent x wishes to delegate a task to agent y. In so doing agent x needs to evaluate the trust it can place in y by considering the different beliefs it has about the motivations of agent y. They claim some trust related beliefs are essential (in x's mental state) to determine the amount of trust to be put in agent y by agent x. This approach considers that the trust relationships of participants in an open network are similar with the social natures of human in the real world. But their model lacks of rational reasoning rules for trust analysis. We propose our model based on this perspective of social trust, and present a formalism of service trust framework supporting analysis and reasoning.

Our social trust approach based on ontology deals with interactions between technology and human social behaviour, and analyzes security requirements as relationships among agents. Our model offers better understanding to the trust relationships in a distributed web services world and will assist participants in making rational operation decisions.

6 References

[1] Castelfranchi. C., Falcone. R., Principles of trust for MAS: cognitive anatomy, social importance, and quantification, *Proceedings of the International Conference of Multi-Agent Systems (ICMAS'98)*, pp. 72-79, Paris, July.

[2] Castelfranchi C., Falcone R., Social Trust: cognitive anatomy, social importance, quantification and dynamics, *Autonomous Agents '98 Workshop on "Deception, Fraud and Trust in Agent Securities",* Minneapolis/St Paul, USA, May 9, pp.35-49.

[3] Castelfranchi, C., and Falcone, R., Social trust: a cognitive approach. *Castelfranchi, C. and Tan, Y.-H. (eds.), Trust and Deception in Virtual Societies.* Kluwer Academic Publishers, pp. 55–90.

[4] Castelfranchi, C. and Falcone, R., Trust is much more than subjective probability: mental components and sources of trust. *Proceedings of the 33rd Hawaii International Conference on System Sciences (online edition)*, Vol. 6. IEEE Computer Society.

[5] Biswas, A., Sen, S. and Debnath. S., Limiting deception in groups of social agents. *Applied Artificial Intelligence Journal* 14(8), 785–797

[6] Sarvapali D. Ramchurn, Dong Huynh and Nicholas. R.Jennings Trust in multi-agent systems. *The Knowledge Engineering Review*, Vol. 19:1, 1–25

[7] McKnight, D. H. and Chervany, N. L. What is Trust? A Conceptual Analysis and An Interdisciplinary Model. *Proceedings of AIS* 2000, 827-833.

[8] Audun Josang Prospectives for Modeling Trust in Information Security. *Australasian Conf. Information Security and Privacy*, 1997: Springer-Verlag,

[9] Audun Josang, Elizabeth Gray and Michael Kinateder Simplification and Analysis of Transitive Trust Networks. *Web Intelligence and Agent Systems, to appear.*

[10] Audun. Jøsang, The right type of trust for distributed systems. *Proc. of the 1996 New Security Paradigms Workshop*, C. Meadows, Ed. ACM, 1996

[11] Lin Liu, Chi Chi-hung, Zhi Jin and Eric Yu Towards A Service Requirements Ontology Based on Knowledge and Intention. *Technical Report*. Knowware Group.

[12] Eric Yu and Lin Liu Modeling Trust for System Design Using the i* Strategic Actors Framework. *In R. Falcone, M. Singh, Y.H. Tan, eds. Trust in Cyber-Societies - Integrating the Human and Artificial Perspectives. LNAI-2246. Springer,2001*. pp.175-194.

[13] Gambetta D. Can We Trust Trust? *Gambetta D, ed. Trust: Making and Breaking Cooperative Relations. Blackwell: Oxford Press*, 1990. 213−237.

[14] Sabater, J. and Sierra, C. 2002 REGRET: a reputation model for gregarious societies. *In Castelfranchi, C. and Johnson, L. (eds.), Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems,* pp. 475–482.