Modeling Trust in Service Oriented Environments

Damjan Kovač, Denis Trček
Laboratorij za E-medije, Fakulteta za računalništvo in informatiko
Tržaška cesta 25, 1000 Ljubljana
E-pošta: damjan.kovac@fri.uni-lj.si, denis.trcek@fri.uni-lj.si

Abstract

Cooperation between entities in e-business environments is based on trust; therefore trust is an essential precondition for every e-business transaction. Trust is too often treated as a hard security mechanism, which can fail when agents act deceitfully. Trust is a personal and subjective phenomenon and has to be treated accordingly. This especially holds true for emerging service oriented architectures (SOAs) that are governed by a set of OASIS WS-* standards where trust needs to gain more attention. We formalize trust by treating it as a soft security mechanism that can be used as a SOA trust management solution.

1. Introduction

Trust is becoming an important aspect in distributed computer environments. When participants of e-business transaction are separated in time and space, no transaction will take place unless the party who moves first possesses satisfactory degree of trust that the party who moves second will indeed honor its commitment.

As stated in the surveys [1, 2] trust is a complex subject relating to the belief in the honesty, truthfulness, competence, and reliability of the trusted person or service. Trust is composed of many different attributes: reliability, dependability, honesty, truthfulness, security, competence, and timeliness, which may have to be considered depending on the context in which trust is specified. Although there are several definitions of trust [3, 4], Grandison and Słomans [1] seems to be the most satisfactory one - trust is the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context. Ultimately, trust is a personal and subjective phenomenon based on various factors.

Many researchers simply use and assume the definition of trust as a security mechanism that mostly covers different authentication and authorization techniques [5] (e.g., access control mechanisms). Information security aims to preserve the CIA properties (Confidentiality, Integrity, and Availability) [6] of information assets within a specific domain. CIA can be assured using different cryptographic protocols and mechanisms [7]. This can be characterized as traditional security mechanisms that protect assets and resources from malicious users, by restricting access to authorized users. However, in many situations we have to protect ourselves from those who offer resources so that the problem is actually reversed. Information providers, can, for example, act deceitfully by providing false or misleading information. While traditional security mechanisms are unable to protect against such threats, trust and reputation systems can. The difference between these two approaches to security domain can be described using the terms hard security for traditional mechanisms (e.g., authentication and access control) and soft security for social control mechanisms (e.g., trust and reputation systems) [4, 8].

The concept of trust also incorporates trust management systems that include collecting the information necessary to establish a trust relationship and dynamically monitor and adjust the existing relationships [9, 2]. They include a centralised and a distributed implementation approaches [8]. The concept of reputation is in close relation to trust. It can be formulated as a collective measure of trustworthiness based on the recommendations from the members of the social network. These recommendations should be based on first-hand experience only. An individual's subjective trust is a combination of reputation and personal experience. In the case of absence of personal experience, trust is based on reputation, which incorporates reputation management systems as well. A survey [2] outlines some traditional trust management systems and gives basic classification to qualitative and quantitative trust models.

Nowadays a new distributed software paradigm is emerging – service oriented architectures (SOAs), a new dimension of Global Computing. SOAs promise a new generation of information systems applications based on a new set of standards (e.g., XML, SOAP, WSDL, UDDI) for enabling self-describing interoperable Web services. A reasonable technology for implementing SOAs are Web services as atomic building blocks. Therefore, the trust of SOA as a soft security mechanism has to be considered. OASIS WS-*
standards (WS-Security, WS-Trust) that ensure hard security mechanisms (e.g., securing SOAP messages) of service oriented applications already exist. The trust as a soft security social mechanism is not covered (actually, no SOA standard treats trust from the social perspective).

Our main contribution in this paper is a formally defined qualitative trust model that covers social mechanism and can be incorporated in an SOA infrastructure. Section 2 formally defines trust model from the qualitative perspective. TODO. Section 4 concludes the paper with some final remarks.

## 2. Modeling trust

A trust model that is based on the social component of the agents is presented in this section. The agent can be an abstract entity in the distributed network and can represent various entities: an ordinary user, a service, a computer, etc. Trust is a measurable quantity and is computed as a function that depends on different trust factors:

- **Time dynamics**: interactions between agents take place over time. An agent’s trust towards the target entity is a dynamic relation that is changing over time. The time is obligatory variable in the trust model. Time dimension is denoted with partially ordered discrete set \( T \) of time values \( t_i \in T \) (\( i \in \mathbb{N}, t_i \leq t_{i+1} \)).

- **Context**: a trust relationship between two agents has a scope, meaning that it applies to a specific purpose or domain, such as “being authentic” in the case of an agent’s trust in a cryptographic key. The context defines an interaction scope between entities. The context is a set of specific trust purposes (e.g., propositions, facts, situations) denoted with discrete set \( \Phi \) of propositions \( \varphi \in \Phi \).

- **Past experience (knowledge)**: the present trust between agents \( a \) and \( b \) is based on the experiences that agent \( a \) has about \( b \); they are a result of past interactions. If no experiences exist, trust can be derived from reputation that the target agent has in the whole community. The knowledge (e.g., overall past trust experience) is denoted by \( \lambda_{a,b}(\varphi) \) and can be considered as the quantity of experiences that agent \( a \) gained from the past interactions with the selected target agent \( b \) in the context \( \varphi \).

First, some basic definitions of trust have to be discussed.

### Definition 1 (Trust relation). A trust relationship between agents \( a \) and \( b \) from the set of agents \( \mathcal{A} \) and is expressed as a binary relation \( R \subseteq \mathcal{A} \times \mathcal{A} \). The trust relation \( (a, b) \in R \) (i.e., \( aRb \)) represents a directed link between agent \( a \) and agent \( b \) in a community that is presented as a directed trust graph \( G(\mathcal{A}, R) \) in time \( t \) and context \( \varphi \).

When focusing on actual trust relationships within a specific context, some properties of trust relation \( R \) have to be pointed out: (i) \( R \) is generally not reflexive as there exists some context where an agent cannot trust him/herself (e.g., performing a surgery), (ii) \( R \) is not symmetric as an agent may trust another while the opposite might not be true, (iii) \( R \) is generally not transitive.

### Definition 2 (Trust degree). An element \( e \in R \) has a degree of trust from a domain set of possible trust values \( \mathcal{D} \). The degree of trust is defined by the function \( \omega : R \times T \times \Phi \mapsto \mathcal{D} \cup \{-\infty\} \). If \( e = (a,b) \) then \( \omega(e) \) can be expressed as \( \omega_{a,b}(t, \varphi) \), \( t \in T, \varphi \in \Phi \) and denotes agent’s subjective trust attitude (opinion) towards agent \( b \) in time \( t \) and context \( \varphi \).

### Definition 3 (Trust domain). The domain set of trust values \( \mathcal{D} \) has a binary partial ordering relation *greater than* (\( \geq \)) that indicates that greater elements mean more trust. A partial order \( (\mathcal{D}, \geq) \) is a complete lattice, which denotes that every subset \( X \subseteq \mathcal{D} \) has the least upper bound \( t \) (supremum) and the greatest lower bound \( \bot \) (infimum).

The trust relations in the social network of agents are represented as a trust graph \( G \). It represents a directed graph with agents as nodes and edges as trust relations among them. The edges are directed and weighted according to a trust opinion \( \omega \). If an edge indicates a trust attitude \( \omega_{a,b} \) of agent \( a \) towards agent \( b \), it is directed from node \( a \) to node \( b \) with the weight \( \omega_{a,b} \).

### Definition 4 (Trust matrix). Social interactions in a social network of \( n = |\mathcal{A}| \) agents is represented with a trust matrix \( M \), where elements \( \omega_{i,j} \in \mathcal{D} \cup \{-\infty\} \) indicate a trust relation \( R \) of \( i \)-th agent towards \( j \)-th agent. If a relation is not defined, it is indicated as \(-\infty\).

The matrix represents trust in a social network at a specific time \( t \in T \) and specific context \( \varphi \in \Phi \) denoted as \( M(t, \varphi) \):

\[
M(t, \varphi) = \begin{bmatrix}
\omega_{1,1} & \omega_{1,2} & \cdots & \omega_{1,n} \\
\omega_{2,1} & \omega_{2,2} & \cdots & \omega_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
\omega_{n,1} & \omega_{n,2} & \cdots & \omega_{n,n}
\end{bmatrix}
\]  

(1)

The rows of the trust matrix represent a certain agent’s trust towards other agents and can be expressed as the image of agent \( a \) denoted by \( a.R = \{ b \in A \mid (a, b) \in R \} \). The columns of the trust matrix represent the trust of the whole social network.

\footnote{Note that \( \omega_{i,j} = -\infty \) does not imply \( \omega_{j,i} = -\infty \).}
regarding to a particular agent; the columns can also be called trust vectors. They can be defined as a pre-image of agent b denoted by $R.b = \{a \in A \mid (a, b) \in R\}$.

The dimension of past experiences represents a derived trust value of all past interactions between agents in a specific context. It can be computed as weighted sum of past interactions where close past interactions have more importance than the distant ones. We use appropriate weight factor. The operation of weighted sum can be performed on set $\mathcal{R}$, so a specific mapping is required from $\mathcal{D}$ to $\mathcal{R}$ and vice versa.

Definition 5 (Past experience). The past experience that $i$-th agent has towards $j$-th agent in the context $\varphi$ can be expressed as weighted sum $\sigma_{i,j} \in R$ of past opinions $\omega_{i,j}$:

$$\sigma_{i,j}(t', \varphi) = \frac{\sum_{t \leq t'} e^{-\alpha(t'-t)} f_D(\sigma_{i,j}(t, \varphi))}{m}, \quad \omega_{i,j} \neq -\infty, \quad \omega_{i,j} = -\infty$$

(2)

where $t' \in T$ is time constant (i.e. observation time), $e^{-\alpha(t'-t)}$ is a weight factor with $\alpha \in [0, 1]$, $f_D : \mathcal{D} \cup \{-\infty\} \mapsto R$ is an injective mapping, $i, j \in [1, |A|]$, $m$ is the number of summed opinions. Past experience $\lambda_{i,j}(\varphi) \in \mathcal{D}$ is denoted as $\lambda_{i,j}(\varphi) = f_D(\sigma_{i,j}(t', \varphi))$ where $f_D^* : \mathcal{R} \mapsto \mathcal{D} \cup \{-\infty\}$.

As we have past experience trust values of all agents, a new trust matrix $M'(t', \varphi) = [\lambda_{i,j}]_{i,j \leq n}$ can be expressed. The new matrix is dependent only on the context dimension $\Phi$: the time dimension $T$ is fixed to the observation time $t'$. It should be emphasized that operations on matrices $M$ are not the same as those in ordinary linear algebra. Trust operations are used instead. A trust operation is a mapping function $\mathcal{D} \cup \{-\infty\} \times \mathcal{D} \cup \{-\infty\} \mapsto \mathcal{D} \cup \{-\infty\}$.

Definition 6 (Trust operation). A trust operation is denoted by an expression $\tau_{i,k}(\varphi) = \lambda_{i,k}(\varphi) \triangledown \lambda_{i,k}(\varphi)$ where $\tau, \lambda \in \mathcal{D} \cup \{-\infty\}, i, j, k \in [1, |A|], \triangledown \in \mathcal{O}$ is an operator that models trust of $i$-th agent, while $\lambda_{i,k}(\varphi)$ and $\lambda_{j,k}(\varphi)$ denote past experience trust values of $i$-th and $j$-th agent towards $k$-th agent in context $\varphi$.

We formalize our trust model as an abstract one being the set $\mathcal{D}$ of past interactions that define trust rules for accessing an agent’s services or resources.

2.1. Qualitative trust model

One of the advantages of the qualitative approach to modeling trust is the fact that it is more comfortable for human beings to think about trust in qualitative terms from discrete set with a reasonably small number of elements. In any case, this set as well as the semantics associated with its individual elements are assumed to be universally known and agreed upon.

Preposition 1 (Trust domain). The domain set $\mathcal{D}$ contains five qualitative elements: distrusted (d), partially distrusted (pd), undecided (u), partially trusted (pt), and trusted (t). The set is $\mathcal{D} = \{d, pd, u, pt, t\}$.

Preposition 2 (Mapping functions). The injective function $f_D : \mathcal{D} \cup \{-\infty\} \mapsto [-1, 1]$ is defined as: $f_D(d) = -1, f_D(pd) = -0.5, f_D(u) = 0, f_D(pt) = 0.5, f_D(t) = 1$. The surjective function $f_D^* : [-1, 1] \mapsto \mathcal{D} \cup \{-\infty\}$ is defined as: $f_D^*(\sigma) = d$ where absolute difference $|\sigma - f_D(d)|_{d \in \mathcal{D}}$ is minimal. Note that $f_D(-\infty) = f_D^*(-\infty) = -\infty$.

Preposition 3 (Trust operators). The set of operators is $\mathcal{O} = \{\lambda, \gamma, \sim\}$ where $\lambda$ indicates an optimistic operator, $\gamma$ a pessimistic operator, and $\sim$ a balanced operator. The operators are defined as follows (the symbol "\*" stands for any value from $\mathcal{D}$):

$$\lambda_{i,k} \land \lambda_{j,k} = \max(\lambda_{i,k}, \lambda_{j,k})$$

$$\lambda_{i,k} \triangledown \lambda_{j,k} = \min(\lambda_{i,k}, \lambda_{j,k})$$

$$\lambda_{i,k} \triangledown \lambda_{k,j} = \min(\lambda_{i,k}, \lambda_{j,k})$$

$$\lambda_{i,k} \triangledown \lambda_{k,j} = \max(\lambda_{i,k}, \lambda_{j,k})$$

The precedence of operators is equal and any specific precedence has to be enforced by the use of parentheses. This is important when applying operators on trust vectors. The computation algorithm $\Gamma$ is dependant on trust operators. Computation of trust on a particular trust vector of a past experience trust matrix $M'(t', \varphi)$ is a dynamic function that includes $i$-th own experience and experiences of the whole society towards target $j$-th agent. The computed trust value $\tau_{i,j}(\varphi)$ in a certain trust vector $\lambda_{i,j} = const$ is indicated as $tvec_{i,j}^c(\varphi)$ and is defined as (where $\lambda_{i,j} \neq -\infty$):

$$tvec_{i,j}^c(\varphi) = (\triangledown(\lambda_{i,j} \triangledown \lambda_{1,j} \triangledown \lambda_{2,j} \ldots) \triangledown \lambda_{n,j})$$

(3)
When no past experience between \(i\)-th and \(j\)-th agent exists (e.g., \(\lambda_{i,j} = -\infty\)) the trust value cannot be computed and the reputation of \(j\)-th agent is used instead. The reputation of \(j\)-th agent in the context \(\varphi\) is indicated as \(\text{rep}_{j}^{\text{op}}(\varphi)\) and is defined as \((k \neq i)\):

\[
\text{rep}_{j}^{\text{op}}(\varphi) = (\ldots(\lambda_{j,j} \text{ op } \lambda_{2,j})\ldots \text{ op } \lambda_{k,j})\ldots) \text{ op } \lambda_{n,j} = \text{ op}_{k=1,k\neq i}^{n}((\lambda_{k,j})
\]

We can now summarize a trust computation algorithm where a computed trust value is the essential decision information for agent \(a\) to or not to interact with agent \(b\) in a specific context \(\varphi\).

**Preposition 4 (Algorithm).** The computed trust of agent \(a\) towards agent \(b\) for a selected operator \(\text{op} \in \mathcal{O}\) in the context \(\varphi \in \Phi\) is formulated as \((a, b \in \mathcal{A})\):

\[
\text{trust}_{a,b}^{\text{op}}(\varphi) = \begin{cases} 
\max(\lambda_{a,b}(\varphi), \text{trust}_{a,b}^{\text{op}}(\varphi)), & \lambda_{a,b}(\varphi) \neq -\infty \\
\text{rep}_{a,b}^{\text{op}}(\varphi), & \lambda_{a,b}(\varphi) = -\infty 
\end{cases}
\]

The set \(\mathcal{P}\) contains interaction trust policies. The trust policy is similar to the security policy where the owner of information resources authorizes certain users to perform specific actions based on user’s credentials. Compliance checking with trust policy is based on the agent’s interaction history that belongs to the specific context. The context can be formulated in different manners (e.g., Kripke model [10], event structures [11]) to carry proper information (e.g., events, states) about each interaction. A proper declarative language (e.g., variants of temporal logic) is used for specifying trust policies and reasoning. The actual implementation of presented trust model could include a mapping between a declarative language and WS-Policy [12] specification.

We have already implemented presented qualitative trust model as trust management system prototype \textit{trustGuard} that is built on the WCF\(^2\) framework using centralized approach. It uses trust Web service (TWS) as an interface to the trust engine. TWS offers insert and query operations with suitable trust primitives defined using XML Schema [13].

### 3. Conclusion

The topic of trust in open computer environments and their applications attracts considerable attention. We have presented a formal definition of an abstract trust model \(TM = (G, D, O, \Omega, F, P)\) from the qualitative perspective. It is based on the temporal and context dimension of cooperating agents and can be characterized as a soft security mechanism. Formal trust model is a foundation for a trust engine of the SOA-based applications. It is implemented as Web service with appropriate operations.

Our future work will focus on the enhancements of the proposed trust model with the agent’s additional social dimensions (e.g., utility, risk) and different trust computation models for the composed Web services (e.g., BPEL processes).

### References


\(^2\)Windows Communication Foundation