Trust Management for Pervasive Computing Environments

Denis Trček, Member, IEEE

Abstract—Trust is essential for further and wider acceptance of contemporary e-services. It was first addressed almost thirty years ago in Trusted Computer System Evaluation Criteria standard by the US DoD. But this and other proposed approaches of that period were actually solving security. Roughly some ten years ago, methodologies followed that addressed trust phenomenon at its core, and they were based on Bayesian statistics and its derivatives, while some approaches were based on game theory. However, trust is a manifestation of judgment and reasoning processes. It has to be dealt with in accordance with this fact and adequately supported in cyber environment. On the basis of the results in the field of psychology and our own findings, a methodology called qualitative algebra has been developed, which deals with so far overlooked elements of trust phenomenon. It complements existing methodologies and provides a basis for a practical technical solution that supports management of trust in contemporary computing environments. Such solution is also presented at the end of this paper.

Keywords—internet security, trust management, multi-agent systems, reasoning and judgment, modeling and simulation, qualitative algebra

I. INTRODUCTION

TRUST is an important phenomenon that forms the basis for many of our everyday decisions. Cyber space is no exception - the more sensitive an interaction in terms of security, privacy or safety is, the more trust there has to exist for an entity to engage into an interaction. The importance of trust is evident also to the highest ranking officials in the EU Commission that are stating that "there is not yet enough trust in the Net" [1].

Before going into methodological details it is necessary to give the basic definitions first. According to the Cambridge Advanced Learner’s Dictionary, trust is a belief or confidence in the honesty, goodness, skill or safety of a person, organization or thing. For trust management in e-environments, this definition is not sufficient. A better definition is the one provided by Denning at the beginning of the nineties [2], when trust started to be more and more exposed in relation to security in information systems (IS). She vividly concluded that trust is not a property of an entity or a system, but is an assessment. Such assessment is driven by experience, it is shared through a network of people interactions and it is continually remade each time the system is used.

And what is reputation? According to the Cambridge Advanced Learner’s Dictionary, reputation is the opinion that people in general have about someone or something, or how much respect or admiration someone or something receives.

D. Trček is head of Laboratory of e-media at Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia / EU, e-mail: (denis.trcek@fri.uni-lj.si).
At the turn of the century, EU funded projects followed that targeted trust. These attempts were already closer to addressing user behavior and the essence of trust, but many can be still characterized as largely security related technologies - two of them follow next. ITrust was a forum for cross-disciplinary investigation of the application of trust as a means of establishing security and confidence in the global computing infrastructure, where trust was recognized to be a crucial enabler for meaningful and mutually beneficial interactions [7]. And TrustCOM was a framework for trust, security and contract management in dynamic virtual organizations. It was intended to be an open source reference implementation that builds on public specifications [8].

Getting now to the theoretical basis, trust in computing environments is most often treated on the basis of Bayes theorem as the starting point. The theorem states that the environments is most often treated on the basis of Bayes builds on public specifications [8].

A generalized Bayes theorem, the Dempster-Shaffer theory of evidence, extends the classical concept of probability, where a probability \( p \) of stochastic event \( x \), i.e. \( p(x) \), and probability \( p \) of its complement \( \bar{x} \), i.e. \( p(\bar{x}) \), sum up to 1. It does this by introducing uncertainty, meaning that \( p(x) + p(\bar{x}) < 1 \). The theory serves as a basis for subjective algebra, developed by Jøsang that is also used in computational trust management [10]. This algebra defines a set of possible states, a frame of discernment \( \Theta \). Within \( \Theta \), exactly one state is assumed to be true at any time. So if a frame of discernment is given by atomic states \( x_1 \) and \( x_2 \), and a compound state \( x_3 = x_1 \land x_2 \), which means that \( \Theta = x_1, x_2, x_1 \lor x_2, x_3 \), then, the belief mass is assigned to every state and in case of, e.g. \( x_3 \) it is interpreted as the belief that either \( x_1 \) or \( x_2 \) is true (an observer cannot determine the exact sub state that is true). Belief mass serves as a basis for belief function, which is interpreted as a total belief that a particular state is true, be it atomic or compound. This gives a possibility for rigorous formal treatment on a mathematically sound basis: In addition to traditional logical operators, subjective algebra introduces new operators like recommendation and consensus, and trust is modeled with a triplet \((b, d, u)\), where \( b \) stands for belief, \( d \) for disbelief and \( u \) for uncertainty. Each of those elements obtains its continuous values from a closed interval \([0, 1]\), such that \( b + d + u = 1 \).

The drawbacks of the above methodologies will be discussed in the next section.

III. QUALITATIVE ALGEBRA

As stated in the introduction, the basis for methodology presented in this section is the research done in the area of psychology that provides an additional useful perspective on trust as a kind of reasoning and judgment process [11], [12]. Taking these works into account, the main factors that have to be considered are the following ones (for additional explanations of the above factors and their use for a formalized model that supports trust in computing environments, a reader is referred to [14]):

- time dynamics - agent’s relation towards the object / subject being trusted is certainly a dynamic relation that changes with time;
- rationality and irrationality - an agent’s trust can be driven by rational or irrational factors;
- feed-back dependence - trust is not a result of a completely independent mind, but is influenced by the agent’s environment;
- action binding - trust can serve as a basis for an agent’s actions;
- trust differentiation - trust evolves into various forms because of the linguistic abilities of an entity expressing trust, or its intentions, and because of perception capabilities of a targeting entity.

The above works provide the main guidelines. However, additional reasons that suggest the need for a new, qualitative methodology, are the following (these address the shortcomings of the existing methodologies that are described in the previous section):

1) As to Bayesian statistics based methodologies, subjects have to understand the basic concepts. However, many research results show that users often have problems with basic mathematical concepts like probability (see e.g. [13]). Now even if subjects understand these basic mathematical concepts, very few of them understand advanced concepts that are required by e.g. theory of evidence.

2) Our research indicates that users may prefer qualitative expressions over quantitative ones when trust is in question. This qualitative ordinal scale is likely to consist of five rankings (qualitative descriptions) [14].

These facts led to the need for a complementary method, which will be defined in the rest of this section.

**Definition 1:** Trust is a relationship between agents A and B that may be described as totally trusted, partially trusted, undecided, partially untrusted, and untrusted; it is denoted by \( \omega_{A,B} \), which means agent’s A attitude towards agent B.

Next, the general nature of trust is that it is not reflexive (in certain contexts one may trust himself / herself, in others not), not symmetric (if agent A trusts agent B in a certain context, this gives no basis for automatic conclusion that agent B also trusts agent A), and not transitive (entity A may trust entity B, which in turn may trust entity C, but the latter may not be trusted by A). This suggests that trust is not an easy problem. To enable the analysis and modeling of trust dynamics in social environments trust graphs are introduced. The links of trust graphs are directed and weighted accordingly. If a link denotes trust attitude of agent A towards agent B, the link is directed from A to B. Because graphs can be equivalently presented with matrices, the second basic definition can be given.

**Definition 2:** In a given context \( \Gamma \), propagated trust in social interactions is represented by trust matrix \( M_{\Gamma} \), where elements \( \omega_{i,j} \) denote trust relationships of \( i \)-th agent to-
wards j-th agent, and where its values taken from the set \( \{1/2, 0, -1/2, -1, -\} \). These values denote trusted, partially trusted, undecided, partially untrusted and untrusted relationships. The last symbol, "-", denotes an undefined relation, meaning that an agent is either not aware of existence of another agent, or does not want to disclose its trust toward another agent.

A general form of trust matrix \( M_\Gamma \) of a certain society with \( n \) agents in a given context \( \Gamma \) is as follows:

\[
\begin{bmatrix}
\omega_{1,1} & \omega_{1,2} & \ldots & \omega_{1,n} \\
\omega_{2,1} & \omega_{2,2} & \ldots & \omega_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
\omega_{n,1} & \omega_{n,2} & \ldots & \omega_{n,n}
\end{bmatrix}
\]

An example of a certain society with trust relationships and qualitative weights is given in Fig. 3:

![Fig. 1. An example society that includes a dumb agent (entity 1)](image)

The corresponding matrix is as follows:

\[
\begin{bmatrix}
- & - & - & - \\
1 & 1 & 1 & - \\
-1/2 & 0 & 1 & 1/2 \\
1 & - & 1/2 & 1
\end{bmatrix}
\]

It should be emphasized that trust matrices operations are not the same as those in ordinary linear algebra. Rows represent certain agents trust towards other agents, while columns represent trust of community related to a particular agent (columns will be referred to as trust vectors). Further, an interesting case with this algebra for computing environments is a possibility to include trust about technological components or services. Such component or service is treated as a dumb agent, which is not aware of itself nor its surroundings. The separation of these components or services. Such component or service is treated as a dumb agent, which is not aware of itself nor its surroundings.

In the above matrix, \( p_{i,j} \) states a weight (from the interval \([0,1]\)) that an entity \( i \) is assigning to judgments of entity \( j \). Therefore, rows represent ponders that a certain entity is assigning to judgments of all other entities in a society. In order to keep things simple, this matrix will be left out the rest of the paper.

Now we are able to introduce qualitative operators. These operators are taken from the set \( \{↑, ↓, ↔, ◄, ↓, ⟷, 1, ⊙\} \), and they are defined in detail in table 1, while their description is given below:

- Extreme–optimistic judgment operator, which results in the most positive judgment value in a society; it is denoted by "↑".
- Extreme–pessimistic judgment operator, which results in the most negative judgment value in a society; it is denoted by "↓".
- Centralistic consensus–seeker judgment operator, which results in a value, which is (contrary to the previous operator) "an average rounded away from the 0 value"; it is denoted by "⊙".
- Moderate optimistic judgment operator, which means the expressed judgment is "strengthened" to the next higher level, narrowing the gap towards the aggregated judgment of the rest of community if this is more optimistic than the agent’s trust is (the value changes one level upwards); it is denoted by "↑↑".
- Moderate pessimistic judgment operator, which means the expressed judgment is weakened to the next lower level, narrowing the gap towards the aggregated judgment of the rest of community if this is more pessimistic than the agents trust is (the value changes one level downwards); it is denoted by symbol "↓↓".
- Self–confident judgment operator, which results in the same value after changes are calculated; it is denoted by "−−".

For the calculation of new trust values (and new trust matrix) the following algorithm is defined:

1. Take the first value in a trust matrix.
2. If the value is "-", write again "-", and go to step 6.
3. Calculate the average value in a trust vector by excluding agents own opinion and values marked with "-".
4. Round the obtained average to the nearest possible judgment value from the set of judgment increments \( \{1, 1/2, 0, -1/2, -1\} \).
5. Compute the result \( ω_{i,k}^+ \) according to table 1 by treating the value from step 4 as \( ω_{j,k}^- \), and agents own opinion as \( ω_{i,k}^- \).
6. If there still exist unprocessed values, take the next value from the trust matrix and go to step 2, else stop the procedure.

Now suppose in the example society (see Fig. 1) agent 2 conforms to the optimistic operator, agent 3 to pessimistic operator, while agent 4 is a centralistic consensus seeker, the calculated simulation would be as follows:
calculated values, but only “pure judgments” entered by entity $i$ stands for any value.

$$
\begin{bmatrix}
-1 & -1 & -1 \\
-1 & -1 & -1 \\
-1 & -1 & -1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
-1/2 & 0 & 1 \\
-1/2 & 0 & 1 \\
-1/2 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
1/2 & -1 & 1/2 \\
1/2 & -1 & 1/2 \\
1/2 & -1 & 1/2 \\
\end{bmatrix}
$$

It is important to note that matrices $M_i$ contain non-calculated values, but only “pure judgments” entered by entities. They constitute, so to say, raw data for our calculations that are used by our algebra to support decision making.

The set of important decision making questions (with clearly visible business objectives, including financial consequences) goes as follows:

- By running the simulation on a given society, is the society likely to reach an equilibrium?
- If it does reach an equilibrium, which are the most likely entities that will be trusted by the society and which not?
- How long will it take for the society to reach the most likely state and what state will this be?
- On which part of the society makes most sense to put most efforts to drive the community into a desired state?

Below is an example of trustGuard component that is used for qualitative algebra simulations (see Fig. 3). The parameters were set as follows: Our society consisted of ten agents, of which 40% behaved according to optimistic operator, 20% according to pessimistic operator, and there were 20% opponents and 20% centralists. Further, the initial distribution of trust values in the trust matrix was 20% values denoted by $1$, 20% values denoted by $1/2$, 20% values denoted by $0$, 20% values denoted by $-1/2$, and 20% values denoted by $-1$ (there was no dumb agents). In addition, 30% of agents were allowed to randomly change their operators, and there were five simulation steps between these random changes. After running the situation for a sufficiently long time, we reach an equilibrium of omega values (all undecided), while roughly one third of operators are $\uparrow$, roughly one third $\downarrow$, and the rest $\sim$. Finally, the upper right corner shows society’s attitude toward a specific agent, which is an agent with a bold line around (in our case this is the agent about which all the members of the society are undecided at the end of this simulation).

![Fig. 3. An example of a simulation run with the trustGuard component](image)

Despite the fact that more detailed discussion of the simulation processes exceeds the scope of the paper, an experienced reader can see that this component enables sound simulations by providing e.g. expected values for variables in question, their distribution, etc.

To conclude this section - from the above discussion it follows that what we are dealing with is a non-linear dynamic system. This means that analytic solutions will be mere exceptions and that we will often have to rely on simulations (to search for various heuristics and solutions for typical, reference scenarios, etc.). Despite this, also various interesting theoretical questions can be addressed, and some of them have already been addressed (for more information the reader is referred to [14]).

IV. SUPPORTING QUALITATIVE ALGEBRA IN E-BUSINESS ENVIRONMENTS

Our solution for trust management support is called trustGuard. It consists of two basic structures: the distributed database where trust values (matrices) are stored, and the user interface that accesses this database, performs insertion and
retrieval of these values, and does calculations that are based on qualitative algebra.

The distributed database is implemented on SOA standards, so user interface interacts with these databases through SOAP protocol. For this to happen, the following two primitives are needed. The first one is trustQuery, and the second one is trustReply. These primitives are basically defined using XML schema. But for clarity and conciseness, XML DTD is chosen to present the syntax of trustReply primitive:

```
<!ELEMENT trustResponse (timeStamp,trustMatrix, function?, extension?) >
<!ELEMENT timeStamp ( #PCDATA) >
<!ATTLIST trustMatrix zulu CDATA #REQUIRED >
<!ELEMENT omega (id1, id2, trustAssessment) >
<!ELEMENT id1 (#PCDATA) >
<!ATTLIST id1 URL CDATA #REQUIRED >
<!ELEMENT id2 (#PCDATA) >
<!ATTLIST id2 URL CDATA #REQUIRED >
<!ELEMENT trustAssessment EMPTY >
<!ATTLIST trustAssessment value ([-1 – 0.5]0[0.5]1) "" "" >
<!ELEMENT function (#PCDATA) >
<!ATTLIST function OID CDATA #REQUIRED >
<!ELEMENT extension (#PCDATA) >
```

The generalized time is expressed as Greenwich Mean Time (Zulu) in the form YYYYMMDDHHMMSS, while trust assessment functions are uniquely identified through OIDs [15]. The syntax of trustQuery is similar to the syntax of trustReply, except that there are no trustMatrix elements. The extension element is included and is added in both primitives for future extensions.

Current trustGuard implementation is sufficiently modular to support not only qualitative algebra, but also Bayesian based methodologies, e.g. Jongs’ subjective algebra. As further implementation details exceed the scope of this paper, a reader can find more implementation details in [16].

V. CONCLUSIONS

In the medieval era, Shakespeare advised us to love all, trust a few, and do wrong to none. Later, the famous German poet Goethe, with a strong sense for deep analyses claimed that as soon as one trusted himself (herself), one knew how to live. And recently, prof. H. Smead vividly noted: “When we were young, we didn’t trust anyone over thirty. Now that we’re over thirty, we don’t trust anyone at all”.

It follows from the above sayings that trust is a very sensitive and scarce resource. This especially holds true for e-business environments, where competition is only a few mouse clicks away, while the medium by its nature is not able to provide communication details that are available in an ordinary face to face contacts. Therefore new mechanisms have to be developed and deployed. Further, if users are to be adequately supported when trust management is an issue, the solutions have to be aligned with mental models. These issues have led to the development of qualitative algebra, and they were also the basis for theoretical views, as well as for the practical implementation for pervasive computing environments that is presented in this paper as well.

ACKNOWLEDGMENT

D. Trček wants to thank to Slovene Research Agency ARRS for support of this research with grants J2-9649 and P2-0359 (Pervasive computing).

REFERENCES

etc.). His practical experience with large infrastructures includes establishment of the first IP connection and its management in 1991 (one of the key persons that contributed to establishment of the Slovene Academic Research Network ARNES). He has published over 100 bibliographic items (including journals with SCI JCR impact factor / WoS). He is a member of various international boards and organizations, including the following: editorial board member of the International Journal of Computers and Applications, ACTA Press, 2004-2005, member of program committees of IASTED Software Engineering 05, 06 and 07, 2009 ACM Workshop on Secure Web Services, 2009 IEEE Congress on Services, etc.