Context-aware Trust Evaluation Functions for Dynamic Reconfigurable Systems

Santtu Toivonen
VTT Technical Research
Centre of Finland
P.O.Box 1000, FIN-02044 VTT
Finland
santtu.toivonen@vtt.fi

Gabriele Lenzini
Telematica Instituut
P.O.Box 589, 7500 AN
Enschede
The Netherlands
gabriele.lenzini@telin.nl

Ilkka Uusitalo
VTT Technical Research
Centre of Finland
P.O.Box 1100, FIN-90571 Oulu
Finland
ilkka.uusitalo@vtt.fi

ABSTRACT

We acknowledge the fact that situational details can have impact on the trust that a Trustor assigns to some Trustee. Motivated by that, we discuss and formalize functions for determining context-aware trust. A system implementing such functions takes into account the Trustee's profile realized by what we call quality attributes. Furthermore, the system is aware of some context attributes characterizing additional aspects of the Trustee, of the Trustor, and of the environment around them. These attributes can also have impact on trustor's trust formation process. The trust functions are concretized with running examples throughout the paper.

Keywords

Context-Awareness, Trust Evaluation Functions, Dynamic Reconfigurable Systems

1. INTRODUCTION

Context influences the behavior of an agent on multiple levels. Generally, context is any information characterizing the situation of an entity. An entity, in turn, can be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves [10]. Context-awareness has been recognized in many research areas of information technology, such as information filtering and retrieval [21], service provisioning [24, 36] and communication [26, 11].

Trust is another emerging research subject. Trust is a fundamental factor in human relationships enabling collaboration and cooperation to take place. In Computer Science, Trust Management [6] studies how to establish and to maintain trust relationships among distributed software components such as software agents and web services, and also between users and software components. Trust management is also a way to enhance security and trustworthiness. As such it has been applied for example in the domains of Semantic Web [25], in Global Computing [7], and in Ad Hoc Networks [22].

However, the relationship between context and trust has not received very much attention, apart from some occasional work, such as the ones reported in [28, 33]. This is unfortunate, since such relationship can easily be recognized and its existence justified. The work reported in this paper delves into that topic.

At an abstract level, trust formation can be described with mathematical functions, which take some phenomena as input, and provide a level of trustworthiness as an output. We formalize such functions by putting emphasis especially on the context attributes. More specifically, the "traditional" aspects influencing trust formation, for example reputation and recommendations, are complemented with contextual information. In addition, we concretize the functions via examples.

The rest of the paper is organized as follows. Section 2 summarizes some of the relevant related work. Section 3 introduces the operational framework where trust is evaluated and proposes a distinction between quality attributes and context attributes based on the trust scope. Additionally, Section 3 illustrates the role of context in the trust evaluation process. Section 4 presents the details of the context-aware trust evaluation function. Moreover, it shows how context information can be used to select, among a set of past experiences and a set of recommendations, those that are relevant with regard to the current context. Section 5 exemplifies the use of context in trust evaluation process through an example. Finally, Section 6 concludes the paper and Section 7 points out some of our future work.

2. RELATED WORK

Trust plays a role across many disciplines, including sociology, psychology, economics, political science, history, philosophy, and recently also computer science [12]. For example, Grandison and Sloman discuss properties of varying definitions of trust for Internet applications, and present different trust models dealing with them [13]. They also summarize some well-known trust management tools, such as PolicyMaker [4], KeyNote [5] and REFEREE [8]. Most of these tools are based on the proposal of Blaze *et al.* [6], who first coined the term *trust management.*

Recent approaches to trust management are able to deal with incomplete knowledge and uncertainty (see for example the surveys reported in [12, 13, 17, 29]). Acknowledging uncertainty is particularly suitable when applied to a global computing environment. The trust evaluation functions we study in this paper are part of this global computing approach to trust management. However, unlike other approaches, such as those reported in [1, 2, 15, 17, 19, 20], we do not develop any new algorithms for trust evaluation. Instead, we investigate strategies for enriching traditional

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

trust evaluation functions with the possibility of analyzing contextual information.

We acknowledge several (trust) relationships when studying the context-dependent trustworthiness of a trustee. Therefore, we suggest a solution for using context data to improve the traditional trust establishment, for example when asking for the trustee's reputation. This extends for example the approach reported in [28], in which the trustors are mainly (human) users of some system, and the context typically taken into account is the location/proximity of other users. It also goes beyond [2], where the kind of trust recognized as context-dependent only has to do with roles of human beings (for example, having a different degree of trust to someone acting as a doctor than acting as a car mechanic).

Inspired by [3], we integrate trust evaluation into a wider model where both the relationships and the quality attributes contribute to the evaluation of the composite trustworthiness. Our reputation-based mechanism is intentionally left at the level of templates; various specific computational techniques can be plugged in it. Examples are those using semirings [32], linear functions [35], belief combination functions over paths in the Semantic Web [27], and reputations as described in [22, 16].

In [23], the authors develop a framework to facilitate service selection in the semantic grid by considering reputation information. In the service interrogation phase, users evaluate the reputation of particular services with regard to a certain aggregation of qualities (called context in the paper), to choose a service that meets with their perceptual requirements. In this paper, context is used to refine the trust evaluation process of the qualities of the trustee.

3. OPERATIONAL SCENARIO OF TRUST

Figure 1 depicts our operational scenario of trust. Here, two main actors are involved in the process of trust evaluation: *Trustor* and *Trustee* (see also [13, 14]). Trustor performs the trustworthiness calculation for a certain purpose, called a *trust scope* [1], the object of which is the Trustee.

Definition 1. Trustor *is the entity that calculates the trustworthiness.* Trustee *is the entity whose trustworthiness is calculated. Trustworthiness is modeled with a* trust value. *Trust value expresses the subjective degree to which the Trustor has a justifiable belief that the Trustee will comply the trust scope.*

To evaluate the Trustee's trustworthiness for a certain trust scope, the Trustor analyzes two different kinds of input: *quality attributes* and *context* attributes.

Quality attributes represent the essential data characterizing the Trustee. Without quality attributes, a Trustor has no *a priori* knowledge of the object of trust, and cannot start any trustworthiness determination on rational basis. The only possible decisions in this case are to trust blindly, that is, to adopt an optimistic approach, or to distrust, which means adopting a pessimistic approach [25].

Context attributes represent contextual information that the Trustor may require in addition to the quality attributes, in order to complete the evaluation of the Trustee's trustworthiness. Context attributes may or may not be available at the moment of trustworthiness evaluation. Their absence does not prevent the trustworthiness evaluation process, but can nevertheless affect the result. For example, depending on the scenario, context may express some relevant property characterizing the Trustor, and its impact on the trust evaluation may strongly affect the preliminary result that comes out from the analysis of the quality attributes.

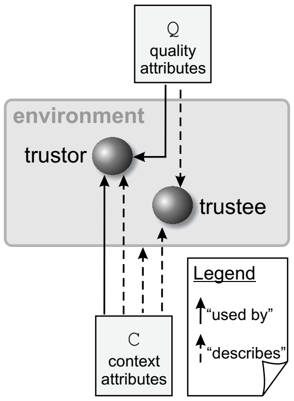


Figure 1: Operational view of trust. The Trustor uses quality attributes and context attributes to decide to what extent it trusts the Trustee. Quality attributes (Q) describe the Trustee's abilities. Context (C) describes surrounding information about the whole scenario constituted by the Trustor, Trustee, and their environment.

The division of one set of attributes into quality and context attributes varies case by case. In this paper, we use the notion of trust scope [1] to deal with the changes affecting this distinction. For instance, suppose that the scope to evaluate a network component is to establish its trustworthiness when it is used in a networked game application. Here, the feature of providing encrypted communication is something that can be understood in connection to the context. Instead, if the same component is judged for trustworthiness when used in a payment application, security features such as encryption are best thought of in connection to the quality attributes.

To conclude this section, we introduce one example of context-depended trust scenarios. It will be used later on in the paper when some concepts need to be concretized and discussed.

Example 1 (Messaging). [1]
Alice receives an SMS with the content "We have just won one million euros at the bingo. Cheers Bob". The Trustor is Alice and the Trustee is the message's content.

If the trust scope is to determine the creator/sender of the message (for example, "Is that really Bob who cheers me?"), quality attributes can be the message header (that includes the phone number from where the message originated), and perhaps the network which delivered the message. Context attributes can be the location of the sender, the location of the receiver, the fact that Alice has bought a lottery ticket in the past, the knowledge (say, from local news) that there has been a winner in the bingo, the reputation of the sender ("he likes making jokes" versus "he never makes jokes").

Instead, if the trust scope is to trust the message content as authentic ("Did we really win?"), quality attributes are the

[1] A more extensive version of this example appeared in [33].

message header, the network which delivered the message, the fact that Alice has bought a lottery ticket, the reputation of the sender. Context attributes can be the location of the sender, the location of the receiver, the knowledge that there has been a winner in the bingo. Note that this last attribute has can change significantly Alice's judgement, but the absence of this piece of information does not disrupt the trustworthiness evaluation process.

4. CONTEXT-AWARE TRUST EVALUATION

This section gives a mathematical characterization of the concepts for quality attributes and context attributes illustrated in Figure 1. Moreover, this section characterizes the mathematical structure of a context-aware trust evaluation function in terms of relevant data domains.

4.1 Quality Attributes and Context Attributes

Let us consider the example scenario of trust described in Example 1. Let `Attributes` represent the information that is potentially involved in this instance of the scenario of trust. `Attributes` contains all the potential message headers (here only phone numbers), network names, localities, and reputation information about the sender of the message.

Formally, `Attributes` is a set of typed and structured data over a signature $\Sigma^{(I)} = A_1 \times \ldots \times A_n$, where A_k are types and $I = \langle a_1, \ldots, a_n \rangle$ is an array of type names. A_k's can be atomic or composed, and are not necessarily distinct.

EXAMPLE 2 (MESSAGING CONTINUED).
The set of all potential data in our messaging example are described as follows:

$$\Sigma^{(I)} = \mathtt{number} \times \mathtt{name} \times \mathtt{location} \times \mathtt{location} \times \mathtt{string} \times \mathtt{bool} \times \mathtt{bool}$$

$$I = \left\langle \begin{array}{l} \mathtt{header}, \mathtt{network}, \mathtt{sender\_location}, \mathtt{receiv\_location}, \\ \mathtt{reputation}, \mathtt{bought\_ticket}, \mathtt{winner\_inthe\_news} \end{array} \right\rangle$$

$$\mathtt{Attributes} = \left\{ \begin{array}{l} \langle +390586, \mathit{TrustFone}, \mathit{London}, \mathit{NY}, \text{“\textit{hates jokes}”}, \mathtt{false}, \mathtt{true} \rangle, \\ \langle +316453, \mathit{MalisFone}, \mathit{NY}, \mathit{Dublin}, \text{“\textit{likes jokes}”}, \mathtt{true}, \mathtt{true} \rangle, \\ \ldots \end{array} \right\}$$

As anticipated in Section 3, within an instance of the scenario of trust and in dependence on the trust scope σ, we can identify two different sets of disjunct sub-tuples in `Attributes`:

- the set `Quality` of all *quality attributes*, defined as the set of data over the signature $\Sigma^{(M(\sigma))}$, where $M(\sigma)$ is a sub-tuple of I (written $M(\sigma) \sqsubseteq I$).
- the set `Context` of all *context attributes*, defined as the set of all data whose signature is $\Sigma^{(I-M(\sigma))}$. Here $I - M(\sigma)$ is the tuple obtained by orderly removing the $M(\sigma)$'s items from I.

We assume `Attributes` $=$ `Quality` $\times$ `Context`, without loss of generality.

EXAMPLE 3 (MESSAGING CONTINUED).
The division into sub-tuples for quality attributes and context attributes depends on the trust scope σ. In reference to Example 1, if the trust scope of Alice is to evaluate the trustworthiness of the message as authentic from Bob, quality attributes are the message headers and the network names. Formally:

$$I \sqsupseteq M(\sigma) = \langle \mathtt{header}, \mathtt{network} \rangle$$

$$\Sigma^{M(\sigma)} = \mathtt{number} \times \mathtt{name}$$

$$\mathtt{Quality} = \left\{ \begin{array}{l} \langle +390586, \mathit{TrustFone} \rangle, \\ \langle +316453, \mathit{MalisFone} \rangle, \\ \ldots \end{array} \right\}$$

The remaining attributes define the context:

$$\Sigma^{(I-M(\sigma))} = \mathtt{location} \times \mathtt{location} \times \mathtt{string} \times \mathtt{bool} \times \mathtt{bool}$$

$$I \sqsupseteq I - M(\sigma) = \left\langle \begin{array}{l} \mathtt{sender\_location}, \mathtt{receiv\_location}, \mathtt{reputation}, \\ \mathtt{bought\_ticket}, \mathtt{winner\_inthe\_news} \end{array} \right\rangle$$

$$\mathtt{Context} = \left\{ \begin{array}{l} \langle \mathit{London}, \mathit{NY}, \text{“\textit{hates jokes}”}, \mathtt{false}, \mathtt{true} \rangle, \\ \langle \mathit{NY}, \mathit{Dublin}, \text{“\textit{likes jokes}”}, \mathtt{true}, \mathtt{true} \rangle, \\ \ldots \end{array} \right\}$$

4.2 Trust Evaluation Function

This section describes the structure for the proposed trust evaluation function, taking into account contextual data. We also present a partial implementation, although the generality of our functions allows different implementations as well.

4.2.1 Trust Values

According to Definition 1, trustworthiness is modeled with a value, called *trust value*, which is the final result of a trustworthiness evaluation process. A trust value can be used, in interaction with a risk analysis, to take a decision in the case of uncertainty [18]. In the literature there exist various implementations for trust values. For example in the Subjective logic theory [17, 18, 16] a trust value is a triple (b, d, u) where $b, d, u \in [0, 1]$ and $b + d + u = 1$; they represent an opinion in terms of amount of belief, disbelief, and uncertainty, respectively.

In this paper, we assume a trust value to be a real number in the interval $[0, 1]$. In this case, a trust value is interpreted as a measure of trust: the values 0 and 1 stand for complete distrust and complete trust, respectively. This choice simplifies the exposition of our strategies for trust evaluation, but we claim that our strategy can be adapted to other models for trust values such as that of the Subjective logic.

4.2.2 Basic Trust Evaluation Function

This section describes the basic version of our context-aware trust evaluation function. Later, we show how to cope with reputation and recommendations, which are generally useful capabilities in trust evaluation, context-aware or not. The basic function for context-aware trust evaluation is defined by the following function from attributes to trust values:

$$\mathtt{ctrust}_{S,\sigma} : \mathtt{Quality} \times \mathtt{Context} \rightarrow [0, 1] \qquad (1)$$

Here S is the Trustor, and σ is the trust scope. In this way we underline that a trust evaluation function is subjective to the trustor (see also [13, 14]) and that it depends on the trust scope. Moreover, $\mathtt{ctrust}_{S,\sigma}$ is defined over the data set `Attributes` which, as said in Section 4.1, is split into quality attributes (`Quality`) and context attributes (`Context`) depending on the trust scope σ.

We propose the whole trust evaluation process to be divided into two stages:

- the first stage is any traditional trust determination process;
- the second stage analyzes contextual information to adjust the output of the first stage.

Formally, we propose that the trust function in (1) has the following shape:

$$\texttt{ctrust}_{S,\sigma}(C,Q) \triangleq C \otimes \texttt{trust}_{S,\sigma}(Q)$$

The first stage is depicted by the function $\texttt{trust}_{S,\sigma}(Q)$. This function can be one of the existing procedures coping with trust evaluation, for example the ones specialized for recommendation-based trust management (see for example [17, 22]). $\texttt{trust}_{S,\sigma}(Q)$, when given an array of quality attributes only, returns a trust value.

The second stage is depicted by the operator $\otimes$. This operator iteratively adjusts the trust value provided at the first stage by evaluating piece of context in the array C of context attributes. To construct the "adjusting operator" $\otimes$ we first define, for each data type name a_k, the following entities:

- $p_k : A_k \to \texttt{bool}$, a predicate that expresses some relevant properties over values of type A_k (of name a_k.
- $w_k \in \texttt{Weights}$, a numerical weighting w_k that expresses the impact of the context attributes of type name a_k in process of refinement.

Here, a predicate p will be used to determine whether certain context value c has a positive (`true`) or negative (`false`) influence on the trust tuning/adjusting.

Set `Weights` represents the set of possible weightings. We assume $(\texttt{Weights}, >)$ to be a totally ordered set, with w_0 its minimum element. Weightings are used to increase or decrease the impact of context data during the process of adjusting. The larger[2] the weight, the larger will the tuning effect be. Note that if the weight is large the adjustment can be quite significant: this reflects situation in which that context data (for example the Trustor's location) is considered (by the Trustor) to effect strongly a preliminary trust evaluation based on Trustee's quality attributes only.

The minimum w_0, is devoted to represent the "I do not care" weighting, that is, context attributes of weight w_0 will not have any impact in the process of refinement.

In addition we define two functions

$$\begin{aligned} \texttt{inc} &: \texttt{Weights} \to ([0,1] \to [0,1]) && (2)\\ \texttt{dec} &: \texttt{Weights} \to ([0,1] \to [0,1]) && (3)\end{aligned}$$

for the positive and the negative adjustment of a trust value v, depending on a certain weight w.

Note 1. *Chosen a weighting $w \in \texttt{Weights}$, $\texttt{inc}_w$ and $\texttt{dec}_w$ are the functions of type $[0,1] \to [0,1]$ that given a trust value v return an adjusted (respectively incremented, decremented with regard to the weighting w) trust value v'.*

Definition 2. `inc`, *and* `dec` *are said* well behaving defining functions *if in their own domain:*

1. *For any $w \neq w_0$, $\texttt{inc}_w(v) > v$ and $\texttt{dec}_w(v) < v$, for all $v \in]0,1[$, that is, they represent positive and negative adjustment as expected.*
2. *$\texttt{inc}_{w_0}(v) = \texttt{dec}_{w_0}(v) = v$, that is, weighting w_0 has no impact in the adjustment.*
3. *When $w > w'$, $\texttt{inc}_w(v) > \texttt{inc}_{w'}(v)$ and $\texttt{dec}_w(v) < \texttt{dec}_{w'}(v)$ for all $v \in]0,1[$, that is, the larger the weighting the more the result of the adjustment.*

Note 2. *In items 1. and 3., the exclusion of the points $v = 0, 1$ is due to two main motivations. The first, obvious, is that we cannot go beyond $[0,1]$ when decreasing and increasing. In other words, $\texttt{inc}_w(1) = 1$ and $\texttt{dec}_w(0) = 0$. The latter, concerns the possibility of having $\texttt{inc}_w(0) \geq 0$ and $\texttt{dec}_w(1) \leq 1$; here, because 0 and 1 express complete (dogmatic) belief and complete disbelief, we make the restriction that no change in context can have effect in the trust evaluation.*

Other restrictions over `inc` and `dec` may be required (for example, $\texttt{inc}_w(\texttt{dec}_w(v)) = \texttt{dec}_w(\texttt{inc}_w(v))$, the property of being reciprocally commutative), but here we prefer to define our adjustment functions in the most general way. More specific sub-families of the functions can be introduced case-by-case.

Although we will provide concrete example of adjustment functions in the following section, a comprehensive study over them is beyond the target of this paper and it is left as future work.

Given a trust value v, arrays $C = \langle c_1, \ldots, c_m \rangle$ of context data, $\langle w_1, \ldots, w_m \rangle$ of weights, and $\langle p_1, \ldots, p_m \rangle$ of predicates, the procedure that implements $\otimes$ consistently with certain $\texttt{inc}_w(v)$ and $\texttt{dec}_w(v)$ functions is described by Algorithm 1.

Algorithm 1 Context Tuning

```
procedure $\otimes(C, v)$
    for all $i \leftarrow 1, m$ do
        if $p_k(c_k)$ then $v \leftarrow \texttt{inc}_{w_k}(v)$
        else $v \leftarrow \texttt{dec}_{w_k}(v)$
        end if
    end for
    return $v$
end procedure
```

Example 4.
An instance of our framework can be specified, for example, by setting `Weights` *any interval $[1, N]$ of rational number, with N a fixed constant. In this case $w_0 = 1$. The following family of functions are used to calculate the positive and negative adjustment for a certain weighting w:*

$$\begin{aligned} \texttt{dec}_w(v) &\triangleq v^w \\ \texttt{inc}_w(v) &\triangleq \sqrt[w]{v}\end{aligned}$$

Figure 2 depicts the effect of some example weightings. Note, that `inc` *and* `dec` *are well behaving functions according to Definition 2. Moreover they satisfy the following additional properties:*

4. *$\texttt{inc}_w(\texttt{dec}_w(v)) = v$ and $\texttt{dec}_w(\texttt{inc}_w(v)) = v$, that is, they are mutually commutative;*

[2] When talking about `Weights`, any reference to terms that involve a concept of ordering must be intended with regard to the relation $>$.

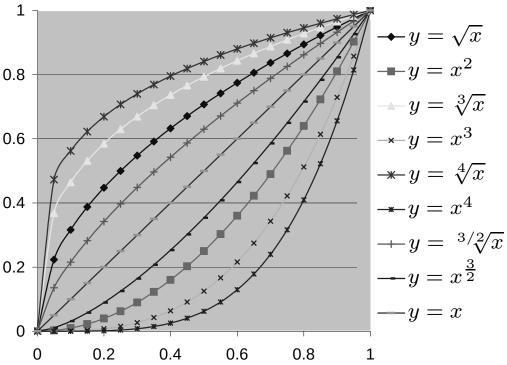


Figure 2: Chart showing the shape of the family of functions $\mathtt{dec}_w(v) = v^w$ ($\mathtt{inc}_w(v) = \sqrt[w]{v}$ resp.) with weight $w \in \{1; \frac{3}{2}; 2; 3; 4\}$

5. *$f_w(g_{w'}(v)) = g_{w'}(f_w(v))$ where $f, g \in \{\mathtt{inc}, \mathtt{dec}\}$, that is, their are order-independent with regard to the context data array.*

Let now suppose to have a trust value $t = 0.7$, and to analyze the context attributes $(c_1; c_2) = (2.2; 2.5)$. The associated weighting are $(w_1, w_2) = (2; \frac{3}{2})$, while the relative predicates are $p_1(c) = p_2(c) = (c > 2.4)$. We apply Algorithm 1 to calculate $(2.2; 2.5) \otimes 0.7$, and we obtain the following trace of execution:

$$\begin{aligned} t' &= \mathtt{dec}_{w1}(0.7) = \mathtt{dec}_2(0.7) \\ &= (0.7)^2 = 0.49 \end{aligned}$$

$$\begin{aligned} t'' &= \mathtt{inc}_{w2}(0.49) = \mathtt{inc}_{\frac{3}{2}}(0.49) \\ &= \sqrt[3/2]{0.49} = 0.56 \end{aligned}$$

The analysis of context attributes has changed a trust value (coming from a first phase) from 0.7 to 0.56.

Additional example functions are briefly discussed in Section 7.

4.2.3 Context Ontology

In the presence of a context ontology which connects the context attributes with each other in an appropriate manner, some reasoning can be made even if assigning the boolean predicate p_k to the context parameter currently under inspection is not possible. The flexibility enables utilising context attributes which do not exactly match the query, but are "close enough" to it [31, 9]. For example, the QoS properties of a network, over which some software component is downloaded, can be described in such ontology (cf. [34]).

Suppose that the current network is not pre-evaluated with regard to its impact on trustworthiness. However, as its neighbors in the ontology are networks which have pre-evaluated trustworthiness values. By using these values as well as their "semantic distance" to the current network, the trustworthiness can be estimated. The Object Match algorithm, outlined in [31], would calculate this semantic distance by taking into account the "upwards cotopy", that is, the distance between the currently investigated concept and a root-concept of the ontology.

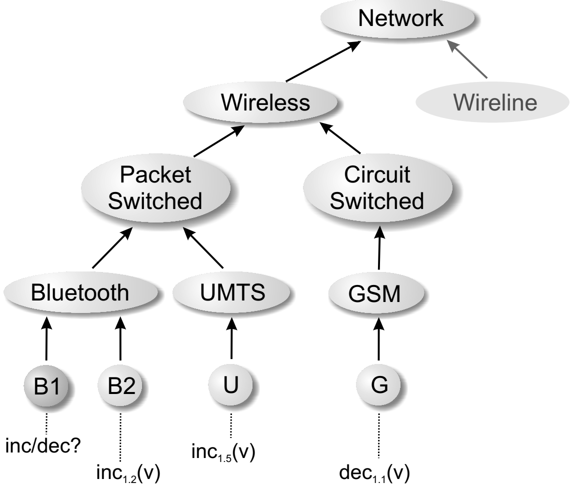


Figure 3: Concepts in the network ontology. The upwards cotopy is calculated as the ratio between the number of shared nodes from the source node and the sink node to the root node, and the total number of nodes from the source and the sink to the root node. For example, in the case of *B1* and *B2*, the numbers are $|Bluetooth, PacketSwitched, Wireless, Network| = 4$ and $|B1, B2, Bluetooth, PacketSwitched, Wireless, Network| = 6$ and the semantic distance between the source and the sink therefore is $\frac{4}{6} \approx 0.67$

Furthermore, the networks are organized in a network ontology, as depicted in Figure 3. Say that the current network *B1* is a bluetooth network, of which there are no pre-evaluated trustworthiness values. However, there exist trustworthiness values of three other networks, which are as follows:

- *B2*, a bluetooth network which would entail $\mathtt{inc}_{1.2}(v)$, semantic distance to *B1* ≈ 0.67
- *U*, a UMTS network which would entail $\mathtt{inc}_{1.5}(v)$, semantic distance to *B1* ≈ 0.43
- *G*, a GSM network which would entail $\mathtt{dec}_{1.1}(v)$, semantic distance to *B1* $= 0.25$

Considering these networks as equal, that is, without taking into account the semantic distance, would entail tuning the trust with $\sqrt[1.2]{\sqrt[1.5]{v^{1.1}}} \approx \mathtt{inc}_{1.64}(v)$. Instead, if the semantic distance is incorporated, the calculation goes as follows: $\sqrt[1.2*0.67]{\sqrt[1.5*0.43]{v^{(1.1*0.25)}}} \approx \mathtt{inc}_{1.89}(v)$. In other words, the trust is increased more, since the kind of network causing the decrement (G) is semantically further away from the current node, and therefore considered less important. This example showed how considering the semantic distance can amplify the increment/decrement effect.

Note that in this example ontology the concepts are organized based on the properties of a network, such as whether the network in question is circuit switched or packet switched. Typically, other details concerning the network, for example its provider, are more important with regard to trust evaluation than its implementation details. That is why the weights assigned for the semantic distance in an ontology such as the one presented in this section should be relatively small. In our approach, the trust related to the the network provider can be considered in terms of reputation and recommendations, both of which will be considered later on in the paper.

4.3 Advanced Trust Evaluation Functions

This section shows how context can be used to complement traditional aspects influencing trust formation. More specifically, we consider reputation and recommendations. Before we can do that, however, we must address the notion of time-line, since it is needed for coping with the history-dependent nature of these topics.

4.3.1 Time Line

We assume a time line for distinguishing between different instances where we apply the trust evaluation procedure. We can generally assume that $\texttt{Time}$ is the set of natural numbers, where $0 \in \texttt{Time}$ is the initial time. With the concept of time we also implicitly assume that the result of a trust evaluation process varies over time. Note that such variation is due to the fact that the input data used by the trust evaluation function changes over time, while the way of reasoning about trust does not. In certain scenarios, even the mechanism of reasoning about trust may change in time, but dealing with this concept of second order dynamism in trust is outside the scope in this paper.

OBSERVATION 1. *In this case the use of time is part of the operational semantics we are giving to our trust evaluation functions. It must not be confused with contextual information "time" that may be used as an input, that is, as part of* $\texttt{Context}$.

If we assume that the trust evaluation happens at time i, we need to bind the time also with the input that is used by the evaluation procedure. Then we indicate with $\texttt{Attributes}^i$ the set of data in the instance of a scenario of trust at evaluation time i

We indicate with $Q^i_\sigma \in Q_\sigma$ the vector of quality attributes that are available for the Trustor at time i. Note that $Q^i_\sigma \sqsubseteq \texttt{Attributes}^i$. We work under the simplified assumption that $Q^0_\sigma = Q^i_\sigma$, for all $i > 0$. This means that the quality attributes do not change along a time line of trust evaluation, unless the Trustee itself is changed. In a more general situation the quality attributes may depend on time. For example, a curriculum vitae of a person may be updated. This assumption allows us to concentrate on contextual aspects and problems. However, should there be a need, some of the techniques here restricted to context attributes, can be applied also to quality attributes. We write $C^i_\sigma \in C_\sigma$ to indicate the state of context at time i.

EXAMPLE 5 (MESSAGING CONTINUED). *In reference to Example 1 and in case of trust scope "Is that really Bob who cheers me?") quality attributes and context attributes at a certain time i are represented by the following tuples:*

$$\begin{array}{l}\texttt{Attributes}^i = \{ \begin{array}{c}\langle +300586, \mathit{MalisFone}, \mathit{NY}, \\ \mathit{Dublin}, \text{"}\mathit{hates\ jokes}\text{"}, \texttt{true}, \texttt{false}\rangle\end{array} \} \\ Q^i_\sigma = \{\langle +390586, \mathit{MalisFone}\rangle\} \\ C^i_\sigma = \{\langle \mathit{NY}, \mathit{Dublin}, \text{"}\mathit{hates\ jokes}\text{"}, \texttt{true}, \texttt{false}\rangle\}\end{array}$$

As a matter of notation, we indicate with $\texttt{ctrust}^i_{S,\sigma}(Q)$ the evaluation of trust performed at time $i \geq 0$:

$$\texttt{ctrust}^i_{S,\sigma}(Q) \triangleq \texttt{ctrust}_{S,\sigma}(Q, C^i_\sigma)$$

The implementation of this function does not change with respect to the one given in the previous section. We only need to bind the evaluation with time i, as follows:

$$\texttt{ctrust}^i_{S,\sigma}(Q) \triangleq C^i \otimes \texttt{trust}^i_{S,\sigma}(Q)$$

here $\texttt{trust}^i_{S,\sigma}(Q)$ represents the result of a context-independent trust evaluation function, applied at time i. Note that although we have assumed Q to remain constant, $\texttt{trust}^i_{S,\sigma}(Q)$ may provide different results along the time. For example, the recommendations may change in the course of time due to the recommenders' new experiences of dealing with the trustee.

4.3.2 Adding Reputations

The next concept we need to consider in trust evaluation is *reputation* [17]. Taking care of the Trustee's reputation means that trust evaluation performed at time $i > 0$ may be affected by past experiences happened at a previous time j, $0 \leq j < i$. Reputation introduces a history-dependent dimension in trust evaluation. We formalize the high-level definition of $\texttt{ctrust}_{S,\sigma}(\_,\_)$ history-dependence by proposing an updated definition of the trust evaluation function, which accepts a trust value as an additional parameter in input:

$$\texttt{ctrust}_S : \texttt{Quality} \times \texttt{Context} \times [0,1] \rightarrow [0,1]$$

We trigger the process of trust evaluation at time $i > 0$ with the following function call:

$$\texttt{ctrust}^i_S(Q) \triangleq \texttt{ctrust}_S(Q, C^i, r^i)$$

where r^i is an appropriate reputation value, available at time i. Here the term "appropriate" means that when we look for a past experience performed in a context that is *compatible* with the one considered at the present time i [2].

We formalize compatibility among two context values c, c' of type a_k, written $c \sim c'$, as the following binary predicate:

$$c \sim c' \iff p_k(c) == p_k(c') \tag{4}$$

Here $==$ means evaluating as the same, that is, $c \sim c'$ if and only if the predicate $p_k(\_)$ returns the same value when applied both to c and c'.

When dealing with an array of context data, we need to calculate their "grade of compatibility", that is, their closeness in terms of the compatibility function $\sim$. To this aim we propose the following function $d(\_)$:

$$\texttt{d}(C, C') \triangleq \sum_{i=k}^{m} \frac{w_k \cdot (c_k \sim c'_k)}{W} \tag{5}$$

where $W = \sum_{k=1}^{m} w_k$. Function (5) measures the weighted and normalized grade of affinity with regard to the predicates we have defined over context type, of two array of context data.

Our selection of a compatible past experience is based on the quest for the experience performed in the past time

M, such that the grade of compatibility with the present context C^i is maximal. In case there exists more than one past experience with this maximum value, the most recent one is chosen. Formally, M is such that:

- $d(C^i, C^M) = \max_{k=1}^{i}\{d(C^i, C^k)\}$
- $\nexists\, M' > M$ such that $d(C^i, C^{M'}) = d(C^i, C^M)$

As a conclusion, we are now able to specify the term r^i, of "appropriate" reputation at time i, as the trust evaluation result of the Trustor S, for scope σ, performed in the most recent past where the context has maximum degree of compatibility with the present one. Formally:

$$r^i = \texttt{ctrust}^M_{S,\sigma}(C)$$

where M is calculated as explained above.

4.3.3 Adding Recommendations

The final concept we need to consider in trust evaluation is *recommendation*. A recommendation is a kind of communicated reputation:

Definition 3 (Recommendation [29]). *A recommendation is an attempt at communicating a party's reputation from one community to another. The parties can be for example human users, devices, software components, or combinations of these.*

Despite the intuitive definition given above, there exists no consensus on the nature of recommendation. In the literature there are two different complementary trends: either a recommendation is or is not a trust value. In the first case, a recommendation is the trust value assessed by the recommender about the Trustee. This option is, for instance, used by Abdul-Rahman and Hailes [2]. A recommender can say, for instance, "in my opinion, c is totally trustworthy" without explicitly providing any proof or data supporting the assessment. In the latter case, a recommendation is any collection of data except a trust value that the recommender possesses about the Trustee. For example, a recommendation can be a log of events describing the recommender's experience with the Trustee [30].

In order to consider the recommendation, the Trustor has to share with its recommender at least a common vision of trust. This statement is implicitly included in Definition 3, where the word "attempt" denotes that the source and target of a recommendation may be incompatible if they belong to different communities [29].

Note 3. *We assume a recommendation to be a trust value.*

The version of the trust evaluation function that considers also recommendations is as follows:

$$\texttt{ctrust}_S : \texttt{Quality} \times \texttt{Context} \times [0,1] \times 2^{[0,1]} \rightarrow [0,1]$$

Here $2^{[0,1]}$ represents the set of recommendations. We trigger the process of trust evaluation at time $i > 0$, with the following function call:

$$\texttt{ctrust}^i_S(Q) \triangleq \texttt{ctrust}_S(Q, C^i, r^i, R^i)$$

where r^i is an appropriate reputation value available at time i, and where R^i is an appropriate set of recommendations available at time i. Again, to obtain "appropriate" reputations, we resort to the context data. Reputations can be filtered by considering the context compatibility. Let us assume to have a certain acceptance grade of compatibility we require in order to consider a reputation to be significant. Here we can use another set of weights, different from the weights we considered when tuning trust. From the set of recommendations R we prune out those which cannot reach the required grade of compatibility.

Let us assume $R = \{(r_u, C_u) |\ u \in S\}$ to be the set of recommendations from a set $\mathcal{R}$ of recommenders. Each recommendation (r, C) carries the context C it relates to. The appropriate set of recommendations we consider in our $\texttt{trust}_{S,\sigma}$ is the filtered set $R^i = \{(r', C') \in \mathcal{R} |\ d(C', C^i) > T\}$, where T represents a compatibility threshold decided by the Trustor. Note that here we are not interested in coping with the set of recommendations and reputations according to the trust management practice, because this problem is assumed to be solved by the function $\texttt{trust}_{S,\sigma}$ we use in the first stage of the evaluation.

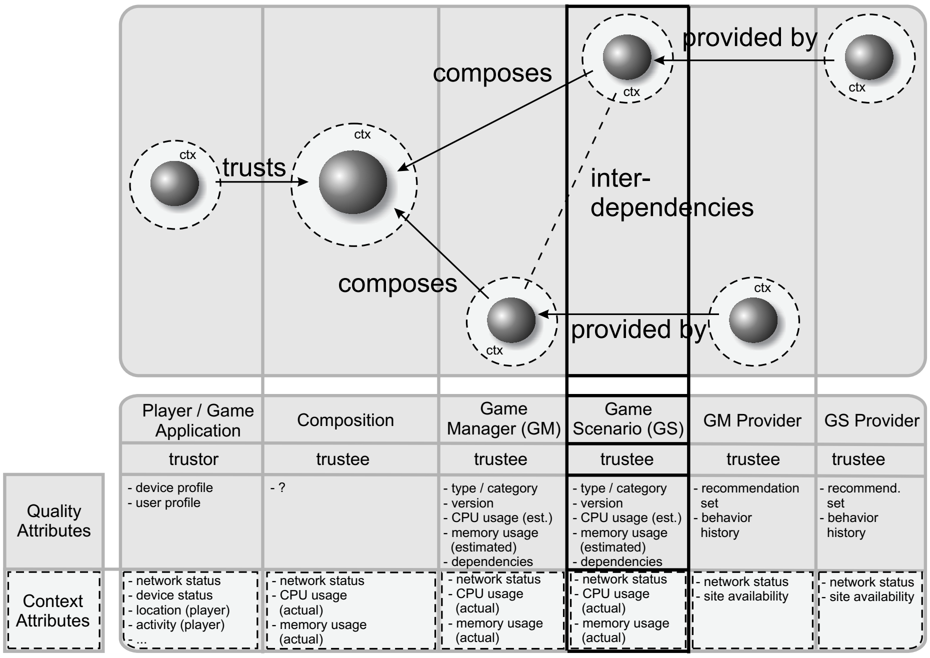
5. EXAMPLE

A game application running on a gaming device is composed by a game manager component (GM) and by one game scenario component (GS). Figure 4 depicts the scenario of a game application composed of these two components. A new game may be composed by downloading new components. Game managers and game scenarios are available on the Internet and they are supplied by different software providers on their Web sites.

Before downloading and installing a new component, the game application checks the hardware and software characteristics of the new game, to evaluate whether the new composition is trustworthy enough or not when running on the current device. This evaluation can include considering both the quality attributes, and the contextual information describing the current situation. It might be the case that the new component is available by different providers or by different mirror sites of one provider. These sites can have varying context attributes such as the current availability. In addition, the sites can have different versions of the needed component(s), which have impact on the interoperability: For example, the `GS_Dungeon_v103` presupposes `GM_v112` or higher, whereas `GS_Dungeon_v102` can manage with `GM_v070` or higher. Furthermore, the different component versions can have varying requirements on the device hard- and software.

We now further concretize the running example by assigning actual values to the context attributes appearing in it. More specifically, we extract two trust scopes (σ_1 and σ_2) for the user/trustor (S). The scopes differ with regard to context. σ_1 has the user on the bus, having access only to a heavily loaded wireless network, and using a small device with limited capabilities (both estimated and actual). σ_2, in contrast, has the user at home, having a broadband access to the Internet, and using a PC with lots of available memory and CPU time.

Furthermore, there are two versions of the Game Scenario components available. Both versions perform the same functionalities and are in that sense applicable in both trust scopes. However, they differ in respects that can be significant in terms of the trust scopes σ_1 and σ_2. Suppose that Game Scenario component version A is large in size, requires



	Player / Game Application	Composition	Game Manager (GM)	Game Scenario (GS)	GM Provider	GS Provider
	trustor	trustee	trustee	trustee	trustee	trustee
Quality Attributes	- device profile - user profile	- ?	- type / category - version - CPU usage (est.) - memory usage (estimated) - dependencies	- type / category - version - CPU usage (est.) - memory usage (estimated) - dependencies	- recommendation set - behavior history	- recommend. set - behavior history
Context Attributes	- network status - device status - location (player) - activity (player) - ...	- network status - CPU usage (actual) - memory usage (actual)	- network status - CPU usage (actual) - memory usage (actual)	- network status - CPU usage (actual) - memory usage (actual)	- network status - site availability	- network status - site availability

Figure 4: Quality attributes and context attributes for a composed game application. For example, in a certain scenario of trust, the trustee can be the Game Scenario (GS) component, and the quality attributes and the context attributes as in the bold bounded column.

a lot of memory and CPU time, its provider has a good reputation based on S's past experience, and the provider is also recommended by a good friend of S. Component version B, in turn, is small in size, requires little memory and CPU. However, its provider is unknown to S and therefore has no reputation history nor recommendations available to S. Say that the initial trust values for the respective components are $t_A : 0.6$ and $t_B : 0.5$ (t_A is a little higher, because A's provider is known by S to have a good reputation and is also recommended to S).

Based on the trust scopes σ_1 and σ_2, S's device can perform the following context-aware trust calculations to the available component versions. In the following we use the definition of `inc` and `dec` given in Example 4:

- Trust scope σ_1
  - Game Scenario component version A
    - Large in size: $\texttt{dec}_2(t)$
    - Requires a lot of memory: $\texttt{dec}_{1.5}(t)$
    - Requires a lot of CPU time: $\texttt{dec}_{1.5}(t)$
    - Good reputation: $\texttt{inc}_{1.25}(t)$
    - Recommended by a friend: $\texttt{inc}_{1.25}(t)$
  - Game Scenario component version B
    - Small in size: $\texttt{inc}_2(t)$
    - Requires little memory: $\texttt{inc}_{1.5}(t)$
    - Requires little CPU time: $\texttt{inc}_{1.5}(t)$
- Trust scope σ_2
  - Game Scenario component version A
    - Large in size: $\texttt{dec}_{1.1}(t)$
    - Requires a lot of memory: $\texttt{dec}_{1.1}(t)$
    - Requires a lot of CPU time: $\texttt{dec}_{1.1}(t)$
    - Good reputation: $\texttt{inc}_{1.5}(t)$
    - Recommended by a friend: $\texttt{inc}_{1.5}(t)$
  - Game Scenario component version B
    - Small in size: $\texttt{inc}_{1.1}(t)$
    - Requires little memory: $\texttt{inc}_{1.1}(t)$
    - Requires little CPU time: $\texttt{inc}_{1.1}(t)$

Based on this information, we can calculate the context-aware trust value. First, for trust scope σ_1 and software version A, we can calculate according to the following steps, starting from trust value t_0, which is 0.6:

$$
\begin{aligned}
t_1 &= (t_0)^2 & &= 0.6^2 & &= 0.36 \\
t_2 &= (t_1)^{1.5} & &= 0.36^{1.5} & &= 0.22 \\
t_3 &= (t_2)^{1.5} & &= 0.22^{1.5} & &= 0.10 \\
t_4 &= \sqrt[1.25]{t_3} & &= \sqrt[1.25]{0.10} & &= 0.16 \\
t_5 &= \sqrt[1.25]{t_4} & &= \sqrt[1.25]{0.16} & &= 0.23
\end{aligned}
$$

So the final value for Game Scenario component A is 0.23. In the same way, component version B in trust scope σ_1 receives the value 0.89. In trust scope σ_2, instead, A receives the value 0.74 and B the value 0.59. In other words, in trust scope σ_1 the component version B is valued over component version A, because it better fits the contextual requirements. In scope σ_2, the valuations for the components are closer to each other, but this time the component version A is valued over B.

This example clearly verifies the hypothesis presented earlier, namely that the weights assigned to the context attributes should be quite small. Here the smallest value assigned for w was 1.1 and the largest 2, and still the trustworthiness values varied between 0.23 and 0.89, therefore consuming a large portion of the scale [0,1].

Another way to draw a line between trust scopes would be to consider the game scenario in one scope, and the whole composite game in another. This way the following situations could be extracted:

Trust scope focusing on the game scenario: The game application is interested in evaluating the trustworthiness of a single piece of software representing the new game scenario. Quality attributes are the names of the component and the provider, version of the component, reputation of the software provider, recommendations from friends on the provider. Context attributes are the actual size of the component being downloaded, the current download speed of the site from where the software is downloaded, the throughput of the network over which the software is going to be downloaded, and the also the hardware characteristics of the game device (its available RAM memory, and the current CPU load).

Trust scope focusing on the composite game: The game application is evaluating the trustworthiness of the composite game as a whole. Quality attributes are all the quality attributes of the components participating in the composition, as well as their providers' quality attributes. In addition, the estimated average CPU and memory usage of GS and GM together and the interdependencies between the versions of the GS and GM components are considered as quality attributes in this example. Context attributes, in turn, are the actual size and resource (CPU and memory) consumption of the downloaded and composed components, and the current hardware characteristics of the game device.

6. CONCLUSIONS

Situational details can have impact on how trustworthy a trustor considers the trustee. These situational details can characterize the trustor, the trustee, and the environment around them. Inspired by this observation, we described and formalized functions for context-aware trustworthiness evaluation. Such functions take into account the individual context attributes, and assign them with values influencing the trustworthiness evaluation process. Depending on the importance of a given context attribute, determined by what we call a trust scope, weights can be applied to amplify or weaken the influence.

Trustee's reputation, that is, the trustor's past observations of the trustee, can further impact the trustworthiness evaluation. We apply the notion of context also to the reputations by emphasizing more the observations that have taken place under similar conditions as where the trustor currently is. Finally, the trustworthiness evaluation can include recommendations from others. There are two relationships between recommendations and context. First, as was the case with reputation, the contextual details at the time when the recommendation was made can be considered and compared with the trustor's current context. Note that considering this is not as straightforward as was the case with reputation, since recommendations come from others, not from the trustor. Secondly, the recommendation content can be context-dependent.

We concretized our formalizations with an example concerning a game application, which is composed out of downloaded components.

7. FUTURE WORK

Our future work includes further refining the trust functions, as well as testing them with real applications. We now present some initial ideas for additional examples of adjusting functions. The first example is an extension of Example 4. We use the same class of functions to define different increment decrement adjustments. The alternative definitions for the positive and the negative adjustment for a weighting $w \in [1, N]$ are defined as follows:

$$\begin{aligned} \mathtt{dec}_w(v) &\triangleq \frac{(v + v^w)}{2} \\ \mathtt{inc}_w(v) &\triangleq \frac{(v + \sqrt[w]{v})}{2} \end{aligned}$$

`inc` and `dec` are well behaving according to Definition 2; moreover, they enjoy the same properties 4. and 5. stated in Example 4.

Another example of families of adjusting functions comes from considering a beams of functions generated by one single "kind" of curve. In this case the weightings are used as amplification/de-amplification factors. For example, if we choose $\mathtt{Weights} = [0, 1]$ a simple example is given as follows:

$$\begin{aligned} \mathtt{dec}_w(v) &\triangleq v + w \\ \mathtt{inc}_w(v) &\triangleq v - w \end{aligned}$$

restricted on $[0, 1]$. Figure 5(A) gives a graphical representation of them.

If we choose $w \in \mathtt{Weights} = [0, \sqrt{2}]$, another family of functions can be defined as follows:

$$\mathtt{dec}_w(v) \triangleq R_{\frac{\pi}{4}} \begin{pmatrix} v' \\ 2wv'(\sqrt{2} - v') \end{pmatrix}$$

$$\mathtt{inc}_w(v') \triangleq R_{\frac{\pi}{4}} \begin{pmatrix} v' \\ 2(-w)v'(\sqrt{2} - v') \end{pmatrix}$$

restricted on the $[0, 1]$. Here $R_{\frac{\pi}{4}}$ is the rotation matrix, and v' is the value corresponding to v in the non-rotated

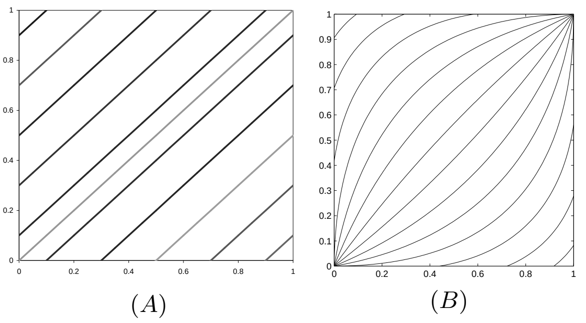


Figure 5: Two beams of functions that can be used to define dec and inc: (A) the beam of strict lines, parallel to y=x, restricted in $[0, 1]$; (B) the beam of parabola $y = 2ax(x - \sqrt{2})$ rotated of anti-clockwise $\pi/4$ and restricted to $[0, 1]$.

coordinated system. Figure 5 (B) shows the graphic of these functions.

We envisage that working with running examples helps us to extract the truly relevant context attributes, as well as give us guidelines on the weights to be assigned to them. In addition, visualizing the trustworthiness evaluation from the end user's perspective should receive some attention. The user should be aware of the characteristics and interrelations of the factors which compose the trustworthiness.

8. ACKNOWLEDGEMENTS

The authors have been supported by the European ITEA project Trust4All. The authors would like to thank the anonymous reviewers for their useful suggestions that brought to an improvement of the paper. S. Toivonen thanks H. Helin for his comments on the context ontology. G. Lenzini thanks A. Tokmakoff for his comments on the whole paper, and I. De la Fuente for her hints on alternative functions.

9. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proc. of the 1997 New Security Paradigms Workshop, Cumbria, UK, 23-26 September 1997*, pages 48–60. ACM and Univ. of Newcastle, ACM Association for Computing Machinery, 1997.

[2] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In I. C. Society, editor, *Proc. of the 334rd Hawaii International Conference on System Sciences (HICSS33), (CD/ROM), Maui, Hawaii, 4-7 January 2000*, volume 6 of *HICSS Digital Library*, pages 1–9. IEEE Computer Society, 2000.

[3] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *Proc. of the 4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 1005–1011. ACM, 2005.

[4] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. G. Keromytis. The role of trust management in distributed systems security. In J. Vitek and C. Jensen, editors, *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, State-of-the-Art, pages 185–210. Springer-Verlag, 1999.

[5] M. Blaze, J. Feigenbaum, and A. D. Keromytis. Keynote: Trust management for public-key infrastructures (position paper). In B. Christianson, B.Crispo, W. S. Harbison, and M. Roe, editors, *Proc. of the 6th International Security Protocols Workshop, Cambridge, UK, April 15-17, 1998*, volume 1550 of *LNCS*, pages 59–63. Springer-Verlag, 1999.

[6] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. of the 1996 IEEE Symposium on Security and Privicay, Oakland, CA, USA, 6-8 May 1996*, pages 164–173. IEEE Computer Society, 1996.

[7] M. Carbone, M. Nielsen, and V. Sassone. A formal model for trust in dynamic networks. In *Proc. of the 1st International Conference on Software Engineering and Formal Methods (SEFM 2003), 22-27 September 2003, Brisbane, Australia*, pages 54–59. IEEE Computer Society, 2003.

[8] Y.-H. Chu. REFEREE:trust management for web applications. Technical report, AT&T Research Lab, 1997.

[9] O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, and F. Gandon. Searching the Semantic Web: Approximate Query Processing Based on Ontologies. *IEEE Intelligent Systems*, 21(1):20–27, 2006.

[10] A. K. Dey, D. Salber, and G. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal*, 16((2-4)):97–166, 2001.

[11] F. Espinoza et al. GeoNotes: Social and Navigational Aspects of Location-Based Information Systems. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp 2001)*, pages 2–17. Springer, September/October 2001.

[12] J. A. Golbeck. *Computing and Applying Trust in Web-based Social Networks.* PhD thesis, University of Maryland, Computer Science Department, April 2005.

[13] T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications and Survey, Forth Quarter*, 3(4):2–16, 2000.

[14] T. Grandison and M. Sloman. Specifying and analysing trust for internet applications. In J. L. Monteiro, P. M. C. Swatman, and L. V. Tavares, editors, *Towards The Knowledge Society: eCommerce, eBusiness, and eGovernment, Proc. of the 2nd IFIP Conference on E-Commerce, E-Business (I3E 2002), October 7-9, 2002, Lisbon, Portugal*, volume 233 of *IFIP Conference Proceedings*, pages 145–157. Kluwer, 2002.

[15] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–312, June 2001.

[16] A. Jøsang, L. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems Journal*, 2006. (to appear).

[17] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 2005. (available on line on ScienceDirect) in press.

[18] A. Jøsang and S. L. Presti. Analysing the relationship between risk and trust. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proc. of the 2nd International Conference on Trust Management (iTrust 2004), Oxford, UK, 29 March - 1 April, 2004*, volume 2995 of *LNCS*, pages 135–145. Springer-Verlag, 2004.

[19] K. Krukow, M. Nielsen, and V. Sassone. A framework for concrete reputation-systems. Technical Report RS-05-23, Univ. of Aarhus, Denmark, June 2005.

[20] K. Krukow, M. Nielsen, and V. Sassone. A framework for concrete reputation-systems with applications to history-based access control (extended abstract). In *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS'05), USA, 7-11 November 2005.* ACM Association for Computing Machinery, 2005.

[21] B. Larsen, editor. *Proceedings of the ACM SIGIR*

2005 Workshop on Information Retrieval in Context (IRiX), Copenhagen, Denmark, Aug. 2005. Department of Information Studies, Royal School of Library and Information Science.

[22] J. Liu and V. Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proc. of the 2nd International Conference on Trust Management (iTrust 2004), Oxford, UK, 29 March - 1 April, 2004*, volume 2995 of *LNCS*, pages 48–62. Springer-Verlag, 2004.

[23] S. Majithia, A. S. Ali, O. F. Rana, and D. W. Walker. Reputation-based semantic service discovery. In *Proc. of the 13th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)*. IEEE Computer Society, 2004.

[24] S. Mostefaoui and B. Hirsbrunner. Context aware service provisioning. In *Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS 2004)*, pages 71–80. IEEE, July 2004.

[25] K. O'Hara, H. Alani, Y. Kalfoglou, and N. Shadbolt. Trust strategies for the semantic web. In J. Golbeck, P. A. Bonatti, W. Nejdl, D. Olmedilla, and M. Winslett, editors, *Proc. of the Workshop on Trust, Security, and Reputation on the Semantic Web – hels as part of International Semantic Web Conference (ISWC 2004) , Hiroshima, Japan, November 7, 2004*, volume 127 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[26] J. Pascoe. The stick-e note architecture: extending the interface beyond the user. In *In Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 261–264. ACM Press, 1997.

[27] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In D. Fensel, K. P. Sycara, and J. Mylopoulos, editors, *Proc. of the International Semantic Web Conference (ISWC 2003), Sanibel Island, FL, USA, 20-23 October 2003*, volume 2870 of *LNCS*, pages 351–368. Springer-Verlag, 2003.

[28] P. Robinson and M. Beigl. Trust context spaces: An infrastructure for pervasive security in context-aware environments. In D. Hutter et al., editors, *Security in Pervasive Computing, First International Conference, Boppard, Germany, March 12-14, 2003, Revised Papers*, volume 2802 of *Lecture Notes in Computer Science*, pages 157–172. Springer, 2004.

[29] S. Ruohomaa and L. Kutvonen. Trust management survey. In *Proceedings of the iTrust 3rd International Conference on Trust Management, 23–26, May, 2005, Rocquencourt, France*, volume 3477 of *LNCS*, pages 77–92. Springer-Verlag, May 2005.

[30] V. Shmatikov and C. Talcott. Reputation-based trust management. *Journal of Computer Security*, 13(1):167–190, 2005.

[31] N. Stojanovic et al. Seal: a framework for developing semantic portals. In *K-CAP 2001: Proceedings of the international conference on Knowledge capture*, pages 155–162, New York, NY, 2001. ACM Press.

[32] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In M. Jakobsson and A. Perrig, editors, *Proc. of the 2004 ACM Workshop on Wireless Security, Philadelphia, PA, USA, October 1, 2004*, pages 1–10. ACM, 2004.

[33] S. Toivonen and G. Denker. The impact of context on the trustworthiness of communication: An ontological approach. In J. Golbeck, P. A. Bonatti, W. Nejdl, D. Olmedilla, and M. Winslett, editors, *Proc. of the Workshop on Trust, Security, and Reputation on the Semantic Web – hels as part of International Semantic Web Conference (ISWC 2004) , Hiroshima, Japan, November 7, 2004*, volume 127 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[34] S. Toivonen, H. Helin, M. Laukkanen, and T. Pitkäranta. Context-sensitive conversation patterns for agents in wireless environments. In S. K. Mostéfaoui, Z. Maamar, and O. Rana, editors, *Proceedings of the 1st International Workshop on Ubiquitous Computing, IWUC 2004, In conjunction with ICEIS 2004, Porto, Portugal, April 2004*, pages 11–17. INSTICC Press, Apr. 2004.

[35] Z. Yan, P. Zhang, and T. Virtanen. Trust evaluation based security solution in ad hoc networks. In *Proc. of the Nordic Workshop on Secure IT Systems (NORDSEC 2003), Gjövik, Norway, 15-17 October 2003*, 2003.

[36] K. Yang and A. Galis. Policy-driven mobile agents for context-aware service in next generation networks. In E. Horlait, T. Magedanz, and R. Glitho, editors, *Mobile Agents For Telecommunication Applications, 5th International Workshop (Mata 2003)*, volume 2881 of *Lecture Notes In Computer Science*, pages 111–120, Marrakesh, Morocco, Oct. 2003. Springer.