Trust Modeling with Context Representation and Generalized Identities

Martin Rehak and Michal Pechoucek

Dept. of Cybernetics and Center for Applied Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague
Technická 2, Prague 6, 166 27 Czech Republic
{mrehak,pechouc}@labe.felk.cvut.cz

Abstract. We present a trust model extension that attempts to relax the assumptions that are currently taken by the majority of existing trust models: (i) proven identity of agents, (ii) repetitive interactions and (iii) similar trusting situations. The proposed approach formalizes the situation (context) and/or trusted agent identity in a multi-dimensional Identity-Context feature space, and attaches the trustworthiness evaluations to individual elements from this metric space, rather than to fixed identity tags (e.g. AIDs, addresses). Trustworthiness of the individual elements of the I-C space can be evaluated using any trust model that supports weighted aggregations and updates, allowing the integration of the mechanism with most existing work. Trust models with the proposed extension are appropriate for deployment in dynamic, ad-hoc and mobile environments, where the agent platform can't guarantee the identity of the agents and where the cryptography-based identity management techniques may be too costly due to the unreliable and costly communication.

1 Introduction

The purpose of our work is to extend the applicability of the agent trust modeling towards open and unmanaged systems common in the ubiquitous computing environments. Instead of devising a novel trust model, we propose an algorithm that can be combined with existing trust models [1,2,3,4] to extend their applicability.

Therefore, we need to address the underlying assumptions of the existing trust models, that assume that *well-identified agents repetitively engage in similar interactions*. An example of a domain where the current trust models are adequate is a supply-chain management, where the agents can be associated to companies or their entities that maintain long-term relationships with repetitive deliveries. A counter-example is a wireless sensor network or a group of Unmanned Aerial Vehicles (UAVs) that are (i) hard to identify and where (ii) the barriers of entry are quite low thanks to the inherent openness of the system. Agents labeled as untrustworthy can therefore easily change their identity and enter the system with a new, uncompromised one. Furthermore, possible interactions are diverse and depend heavily on the context (e.g. sensor performance that varies between day and night).

1.1   Assumptions of the Existing Trust Models

Most existing trust model are based on an assumption of the **proven identity**: it is assumed that the agents can't change their identity during their lifetime and that they can't have multiple identities at once. This assumption is acceptable in purely multi-agent systems, where the agent platform manages and oversees the agents, but fails in case of open systems (featuring mobile agents and ad-hoc networks), where the agents are distributed across independent platforms and can join or leave the system at their own will. Cryptography-based identification methods [5] can address the problem rather efficiently, but the need to verify the chain of identity to the mutually recognized authority may pose severe problems in a ubiquitous computing context, where the resources and network connectivity are scarce. In such environments, the delegation of the identity management to a dedicated infrastructure is costly, possibly leaving the system designers with a choice between secure inflexible system, or an unsecured, but more robust open system. This contribution suggests how to complement the existing methods and addresses this gap.

Identity problem is closely related with a **"first time offender"** problem. Current trust models concentrate on managing ongoing relationships with repetitive interactions. While the reputation (i.e. witness reputation, trust communication) and social dimension [6] of certain models alleviate the situation, they can't by any means protect the open system against the agents that connect, build-up their trustworthiness and use it to defect when the stakes are higher.

Last, but not least, existing trust models often neglect the problem of **context** (e.g. situation) – they are mostly based on the assumption that the interactions with a given provider are similar and that the previously acquired experience is relevant for future interactions. We argue that the inability to take the context into account limits the practical use of current trust models in all domains where the agents perform diverse tasks in a highly dynamic environment – this being a precise definition of the ideal environment for deployment of intelligent social agents[7].

In Section 2, we will propose an extension of general trust models that will allow them to consider the situational trust by means of context modeling and to efficiently exploit the similarities between the various situations. In Section 3, we will extend the context representation framework to include an identity of the trusted agent as well and we provide an outline of the algorithm. Section 4 provides an evaluation of our approach in a simulated logistics scenario, before the discussion of the related work and conclusion in Sections 5 and 6.

2   Context Representation

In most existing trust models[3,8,9,10], the trustfulness of an agent X as evaluated by agent A is a weighted aggregation of past observations of X's performance, either direct or indirect, that are available to A. In most models, recent experience, direct observations and coherent reputations provided by trusted agents [11,12] are emphasized, but as these issues are orthogonal to the topic of this paper, we will not explicit them in our notation.

The goal of the formalism presented in this section is to provide an efficient mechanism for situation representation and to show how can be such mechanism integrated with existing trust models. Therefore, we need to capture relevant aspects of the situation and to represent them as a context, a point c_i in the context space $\mathbb{C}$. The context space is a Q-dimensional metric space with one dimension per each represented situation feature, and the metrics $d(c_1, c_2)$ defined on $\mathbb{C}$ describes similarity[1] between the contexts c_1 and c_2 (or rather lack of similarity).

Please note that while the definition of the $\mathbb{C}$ dimensions and distance function are necessarily domain dependent, the model that we propose only imposes minimal requirements on their definition, most notably the properties of the metric space. When these conditions are fulfilled, the algorithm we propose below can be applied.

Example: A trivial example of a trusting decision that is influenced by the context is a territory surveillance by autonomous UAVs. We can consider the territory size as one dimension of the situation, and visibility (day/night/cloudy/clear) as another one (or even as two features). We shall then find good individual metrics - here, a radius of the area to cover (or its logarithm, to emphasize the ratios rather than absolute differences).To represent the visibility, we may consider any standard metrics from the aviation domain or any metrics that appropriate for intended goal.

The distance metrics allows us to integrate the context representation mechanism with the existing trust models. Therefore, we define a set $\mathcal{R}$ of *reference contexts* r_i, the elements of the $\mathbb{C}$ to which we will associate the trustfulness values, e.g. trust model instances. We shall note that while it is highly desirable to share a common $\mathbb{C}$ definition between all the trusting agents in the system, the definition of the set $\mathcal{R}$ is up to each trusting agent and will be probably even different for each partner evaluated by the agent.

The price that we have to pay for context representation is that instead of maintaining a single instance of the trust model structure (representing the general trust) per partner, we shall maintain one instance per partner for each relevant reference context. On the other hand, when we further generalize this approach in Section 3, we will apply the same approach to identity representation as well, potentially *decreasing* the amount of data to hold.

The following relations will present the formulas for trustfulness update and trustfulness aggregation before taking the trusting decision. We will denote as $\Theta_A(X|r_i)$ the

---

[1] Any distance function $d : \mathbb{C} \times \mathbb{C} \to R$ must respect following properties: **non-negativity**:

$$d(c_1, c_2) \geq 0, \tag{1}$$

symmetry:

$$d(c_1, c_2) = d(c_2, c_1), \tag{2}$$

zero distance $\Leftrightarrow$ **identity**:

$$d(c_1, c_2) = 0 \Leftrightarrow c_1 = c_2, \tag{3}$$

triangle inequality:

$$d(c_1, c_3) \leq d(c_1, c_2) + d(c_2, c_3). \tag{4}$$

trustfulness of agent X in the situation represented by reference context r_i, assessed by agent A. To obtain a weight decreasing with the distance, we introduce a domain-dependent weight function $w_i = f(d(c_d, r_i))$, where f is a non-increasing function on $[0, +\infty)$. This function represents the decay of the observation usefulness with increasing distance d of the particular reference context r_i – obviously, it is most useful when its distance $d(c_d, r_i)$ from the reference context is zero. A convenient form of such function can be for example $w_i = e^{-d(c_d, r_i)}$.

In the following, we will use a shorthand definition for the aggregate weight of the p past observations associated with each reference context r_i, denoted W_i^p.

$$W_i^p = \sum_{j <= p} w_i^j \tag{5}$$

Each new **observation** $\tau_A(X|c_o)$ is integrated into the apriori trustfulness evaluation $\Theta_A^p(X|r_i)$ associated with each r_i (if $w_i^p + 1 > 0$) using the following formula:

$$\Theta_A^{p+1}(X|r_i) = WeAg((\Theta_A^p(X|r_i), W_i^p, (\tau_A(X|c_o), w_i^{p+1})) \tag{6}$$

The implementation of the $WeAg()$ operator depends entirely on the trust model used to represent $\Theta_A^{p+1}(X|r_i)$. In a simple example, when the $\Theta_A^p(X|r_i)$ is just a w_i weighted average of all p previous observations, we obtain:

$$\Theta_A^{p+1}(X|r_i) = \frac{W_i^p \cdot \Theta_A^p(X|r_i) + w_i^{p+1} \cdot \tau_A(X|c_o)}{W_i^p + w_i^{p+1}} \tag{7}$$

In order to take a trusting decision in a situation represented by context c_d, we must **query** the model. The result of the query, a shown in Eq. 8, is obtained as a weighted combination of trustfulness associated with relevant (i.e. close) reference contexts from $\mathcal{R}$.

$$\Theta_A(X|c_d) = WeAg_{r_i \in \mathcal{R}}(\Theta_A(X|r_i), w_i) \tag{8}$$

In the simple weighted average case, we obtain:

$$\Theta_A(X|c_d) = \frac{\sum_{r_i \in \mathcal{R}} w_i \cdot \Theta_A(X|r_i)}{\sum_{r_i \in \mathcal{R}} w_i} \tag{9}$$

The query process is illustrated in Fig. 1, where we aggregate the result from four reference contexts in the vicinity, showing the role of the weight w_i as obtained from $d_i = d(c, r_i)$.

2.1 Reference Set

While the above relations are sufficient to explain the update and aggregation process of the trustfulness information associated with the individual reference contexts, they don't address a critical issue of the model: positioning of the reference contexts in the context space.

The method we propose above doesn't require any particular shape of the reference set. The requirements on the reference set maintenance method are relatively severe, as
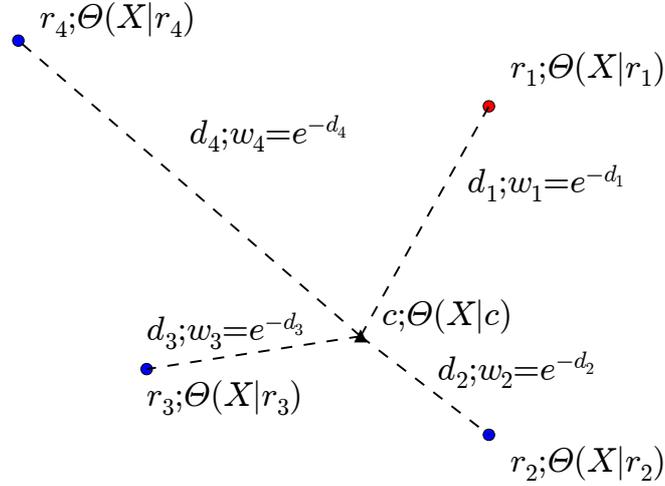
$$r_4; \Theta(X|r_4)$$

$$r_1; \Theta(X|r_1)$$

$$d_4; w_4 = e^{-d_4}$$

$$d_1; w_1 = e^{-d_1}$$

$$d_3; w_3 = e^{-d_3}$$

$$c; \Theta(X|c)$$

$$d_2; w_2 = e^{-d_2}$$

$$r_3; \Theta(X|r_3)$$

$$r_2; \Theta(X|r_2)$$

Fig. 1. Aggregation of the trustfulness of agent X in the situation represented by context c from four reference contexts in the vicinity

we need to handle: (i) real-time reference context placement, without apriori knowledge of the future data distribution, (ii) specificity of the reference contexts positions for each modeled agent and general uncertainty of the distribution of interactions in the space $\mathbb{C}$.

Our current solution is based on Leader-Follower clustering algorithm [13], presented in Fig. 2. The advantages of this particular clustering method are obvious - it is a clustering method that allows on-line approach, without pre-specifying the number of expected clusters beforehand, as is the rule for most other clustering methods. Furthermore, it requires only a single parameter as its input: the cutoff distance of the new context from the closest centroid, when the new context will become a part of the existing cluster. If the new point is farther away, it becomes a centroid of the new cluster.

The biggest disadvantage of this method is that it may easily under or over estimate the number of clusters. In our case, this is not a big problem, as the overestimation results only in maintenance of two or more reference contexts instead of one. Therefore, it is always better to slightly underestimate the cutoff distance, avoiding the data blurring.

3   Including Generalized Identity Management

To represent the uncertain identities, we will extend our context representation formalism to cover the identity as well. Instead of representing the identity as a unique tag, we will decompose the identity into relevant, measurable features, that will form an **identity space** $\mathbb{I}$. The identity space is also a metric space, where the metrics describes lack of similarity between two identities. Please note that the properties of the metrics apply to the formal identities only – the fact that two elements of the space $\mathbb{I}$ are identical (i.e. have 0 distance following 3) doesn't necessarily imply that the respective agents are identical as well – this only means that the features accessible to the model doesn't allow us to differentiate between these identities.

```
class LFClustering:
  def __init__(self, threshold):
    self.centers = []
    self.thresh = 0.5
  def newSample(self, sample):
    closest,dist = self.findClosestCenter(sample)
    if closest != '0' and dist <= self.thresh:
      closest.aggregate(sample, 1, 0.01)
    else:
      self.centers.append(sample)
  def findClosestCenter(self, sample): ...
```

Fig. 2. L-F Clustering algorithm outlined in Python

The fact that two agents with presumably distinct identities can be considered identical by the model provides the protection against changes of identity. When the properties that define the $\mathbb{I}$ are set correctly, the new identity of the agent will fall close to the original one, allowing the evaluating agent to re-use the previous trustfulness data. For example, in sensor networks, we may use the power of the signal or other radio properties of the partner node, combined with the type of services it provides. In the UAV use-case, we may use visual properties of the craft (possibly type identification), radar properties and typical flight patterns to distinguish between evaluated entities.

This approach is (to some extent) also effective against first time offenders. We assume that to take down an open, distributed system working on p2p basis, the attack will be based on similar principles and will consist of relatively large number of similar nodes[2]. Due to the similarity of attacking nodes, the mechanism will treat them as a single identity, or a small number of identities. This means that it will be able to react to the novel threat in a same manner as a biological immune system, where the knowledge gained on a single hostile entity influences the interactions with the other entities of the same type, provided that an efficient reputation-sharing component is a part of the underlying trust model.

Formally, it is preferable to consider the spaces $\mathbb{I}$ and $\mathbb{C}$ as two subspaces of the identity-context space $\mathbb{IC}$, where each vector ix belonging to this space contains all the information associated with an observation or a decision.

For consistency, we will maintain the notation of the reference set $\mathcal{R}$ and reference identity-context r_i, even if their dimension is increased to incorporate the identity subspace dimensions as well.

$$\Theta_A^{p+1}(r_i) = WeAg((\Theta_A^p(r_i), W_i^p, (\tau_A(ix_o), w_i^{p+1})) \tag{10}$$

$$\Theta_A^{p+1}(r_i) = \frac{W_i^p \cdot \Theta_A^p(r_i) + w_i^{p+1} \cdot \tau_A(ix_o)}{W_i^p + w_i^{p+1}} \tag{11}$$

---

[2] In this model, we deliberatively ignore the attacks based on mobile code and environment saturation, that we address in [14].

When we **query** the model to obtain a situation-relevant trustfulness before taking a trusting decision, the current context c_d is determined and the trustfulness is obtained as a weighted combination of trustfulness associated with reference contexts from $\mathcal{R}$.

$$\Theta_A(ix_d) = WeAg_{r_i \in \mathcal{R}}(\Theta_A(X|r_i), w_i) \tag{12}$$

In the weighted average case, we obtain:

$$\Theta_A(ix_d) = \frac{\sum_{r_i \in \mathcal{R}} w_i \cdot \Theta_A(X|r_i)}{\sum_{r_i \in \mathcal{R}} w_i} \tag{13}$$

Before the discussion regarding the implementation of the above-defined relationships, we shall list the domain dependent functions that are used by the model in Table 1.

Table 1. Domain dependent parameters of the model

Code	Description
`identityCx(ag,sit)`	Creates formal identity and context
`distance(x,y)`	$d(x,y)$, Distance between $x, y \in \mathbb{IC}$
`weight(d(rc,x))`	w_i, Weight function
`clustsize`	Max distance for cluster attachment
`treshold`	Min weight for trust update

The implementation of the system is straightforward, outlined in Fig. 3 and Fig. 4. In Fig. 3, we can notice that the update of the trustfulness of individual reference contexts r_i (denoted `rc`) is performed in the same time as the update of the set $\mathcal{R}$ (`rclist`) (either by appending the new reference context id) or by updating the position of the centroid of the cluster to which the new observation was assigned. Therefore, the complexity of the algorithm was increased by a linear factor $|\mathcal{R}|$ (i.e. the size of the `rclist`). We can also note that the context representation is integrated seamlessly, only by means of domain dependent functions `identityCx(ag,sit)` and `distance(rc,id)`. Therefore, removing the context representation altogether or altering its resolution (by lowering/increasing the weight of the context-relative parts of the distance function) will directly reduce the parameter $|\mathcal{R}|$. Such operation can be performed at runtime and can help the agents to manage the computational requirements of their trust model. On the other hand, we can easily use the model with crisp identities, by imposing a trivial distance function in the identity subspace, using the formula $d(x,y) = 0$ iff $x = y$, else $d(x,y) = \inf$. Such definition will effectively split the model into n independent models, where n is the number of evaluated agents[3]. These "submodels" will then function exactly as the context-only mechanism variant presented in Section 2.

---

[3] When applying this approach in the environments with significant number of agents, it is advantageous to replace the linear list `rclist` in Fig. 3 by appropriate partially indexed indexed structure to iterate only over the contexts relative to a particular modeled agent.

```
def newObservation(agent,situat,trustObs):
    # get the identity
    id = identityCx(ag,situat)
    mindist = Infinity
    closest = '0'
    for rc in rclist:
        dist = distance(rc,id)
        # clustering
        if dist < mindist:
            closest = rc
        # update the trustfulness
        wei = weight(dist)
        if wei > treshold:
            rc.trust.update(trustObs,wei)
    # create new reference context if necessary
    if mindist > clustsize:
        rclist.append(id)
        id.trust.update(trustObs,1)
    else:
        closest.updatePosition(id)
```

Fig. 3. Processing of the new observation outlined in Python (simplified)

```
def query(agent,situat):
    id = identityCx(agent,situat)
    result = Trustfulness()
    for rc in rclist:
        wei = weight(distance(rc,id))
        if wei > treshold:
            result.aggregate(rc.trust,wei)
    return result
```

Fig. 4. Processing of the query by the model, (Python, simplified)

While the complexity of the trust model increases by incorporation of the proposed mechanism, the Proposition 1 shows that the memory requirements will in most cases actually *decrease*.

Proposition 1. *Provided that (i) we use the identity representation mechanism as specified by Eq. 10 to Eq. 13, (ii) with the Leader-Follower clustering algorithm as outlined in [13] and Fig. 2, the trust model with uncertain identities will be smaller (in terms of required memory) than the corresponding model with crisp identities. We also assume*

that (iii) both models use the same (possibly none) mechanism for context management and (iv) that the amount of memory used to represent the typical agent identifier in a crisp model (e.g. address, AID,local name) the same or bigger than the size of a single element of the $\mathbb{I}$.

Proof. Using the fact that the size of the structure representing the Θ_A is the same in both cases (iii), as we use the same trust model, and that the size of the associated identities is comparable as well (from (iv)), we only need to compare the number of elements Θ_A in the model. From (ii) and Fig. 2, we can see that we create at most a single centroid per each new identity. Therefore, the size of the model with uncertain identities must be smaller or equal to the crisp model.

The proposition is intuitive from information-theoretic perspective – the trust model based on agent identities is the most detailed model possible, and will therefore constitute an upper bound on the size of any generalizing model. With increasing size of the clusters (and therefore their non-increasing number), the amount of data in the model decreases.

4   Experimental Evaluation: Context Representation

In this section, we will present results of the benchmark performed in a simulated logistics scenario. To evaluate our approach experimentally, we model the trust reasoning of a humanitarian aid organization agent that acquires transportation services from several local transporters after major disaster, and we will enhance it by use of context. We will then compare the performance of the agents who use the baseline model with the performance of the agents using the model enhanced with context representation part as specified in Section 2.

4.1   Context Space Definition

To illustrate the abstract notions of metric space $\mathbb{C}$, we introduce an example of such space for our logistics scenario, where we model each trusting situation (observation or decision) by three parameters: cargo type, cargo size and road quality. Cargo *type* defines the product we transport: medical supplies, food or durable goods. Each cargo type has specific handling requirements – medical supplies are the most sensitive to carry, while the durables require less care. *Size* of the transport is simply a quantity to carry, while the *road quality* represents the quality of the roads to use for transport. It is interesting to note that *type* dimension is discrete, while the *size* and *road quality* are real-valued, but different: one has an absolute scale (size), while the other will be close to 1.

The context space $\mathbb{C}$ is three dimensional, with one discrete dimension and two continuous ones. The next step is a definition of marginal distances d^q for each dimension. In the *type* domain, we place our products on a "sensitivity" scale: medical supplies require most attention: 5, with the food in the middle: 1 and the durables as least sensitive ones, with 0.2 value[4]. Our type distance metrics is defined as follows, using the product properties defined above:

---

[4] Inverting the scale will not change the result thanks to the distance symmetry stated in Eq. 2.

$$d^{type}(c_1, c_2) = |ln(type_1) - ln(type_2)| \tag{14}$$

In the size domain, the metric shall describe the differences between two contracts in terms of their relative size. We propose a measure

$$d^{size}(c_1, c_2) = |ln(size_1) - ln(size_2)| \tag{15}$$

The logarithmic relation captures an intuitive notion of ratio: 10 tons difference between two 20 and 30 ton transports is much more important than the same difference between two shipments of thousands of tons.

We apply the same reasoning for the road quality:

$$d^{road}(c_1, c_2) = |ln(qual_1) - ln(qual_2)| \tag{16}$$

Then we combine the above metrics using a slightly modified (weighted) "Manhattan distance":

$$d(c_1, c_2) = \alpha_1 d^{type}(c_1, c_2) + \alpha_2 d^{size}(c_1, c_2) + \alpha_3 d^{road}(c_1, c_2) \tag{17}$$

4.2 Experimental Setup

In the task allocation problem solved by the agents in our humanitarian logistics scenario, they choose one or more providers (transporters) for each contract and use their trust models to reason about their trustfulness.

In the underlying simulation model, the transporters answer the call for proposals with *bid prices* pr_b based on the nominal transportation cost and profit margins. The *real price*, that includes the cost of the cargo lost during transportation, is derived after the transport from the bid price and transporter *real trustworthiness* Θ. The Θ depends on the same parameters as those that define the $\mathbb{C}$ dimensions. Real price pr_r is determined as $pr_r = \frac{pr_b}{\Theta}$, where

$$\Theta = \Theta_{type} \cdot atan'(price) \cdot atan'(supply) \tag{18}$$

The function $atan'(x)$, used as a sigmoid approximation, is defined as a normalized $arctan$: its range is (x_{inf}, x_{sup}) (both x_{inf}, x_{sup} are in the range set) and x coordinate of its flection point is defined by parameter x_{center}. x_{slope} determines the first derivation - speed of the growth on the domain.

$$atan'(x) = \frac{1 - x_{inf}}{\pi} \cdot arctan(\frac{x_{center} - x}{x_{slope}}) \tag{19}$$

While the provider simulation is a very simple one, it is sufficiently versatile to model the performance of market actors to obtain validation scenarios for our methods.

To evaluate the performance of the evaluated trust models, we introduce the *mean loss*, defined as a difference between the real price pr_r and the bid price pr_b. In the graphs, it is aggregated per all contracts awarded in a single time step. As it is impossible to achieve the zero loss in our scenario, we introduce the optimal choice value, defining the optimal performance of the trust model.

To validate the model independence of the method presented (i.e. the fact that the restrictions placed on Θ modeling are not constraining), we have used two different trust models in our evaluation. The first model, denoted **RNT** in the graphs, we represent the trustfulness in each $\Theta(X|r_i)$ as a time-weighted average of the last N relevant observations. This means that we store N real values in each reference context r_i. The other model, denoted FNT, is a slightly simplified representation described in [10], where each $\Theta(X|r_i)$ is a triangular, asymmetrical fuzzy number.

In Figures 5, 6 and 8, we compare the performance of the trust models without the context representation (denoted **FNT** and **RNT**) with the same models enhanced with context representation, denoted **clusters RNT** and **clusters FNT** that are extended with the above mechanism.

4.3   Influence of Situation Modeling

In the first batch of experiments, we will investigate the influence of the context modeling. We will therefore compare several trust models with and without the context components in the scenarios with increasing level of situation influence on the provider performance. The changes in the performance are modeled by changes of the coefficients in the Eq. 18 and 19.

In the first scenario (Fig. 5), the performance of all the providers is flat over the whole space $\mathbb{C}$ - the outcome of the delegation/contracting is independent of the situation. We may note that the general methods perform slightly better, as their learning process is more efficient, but the differences remain minor.
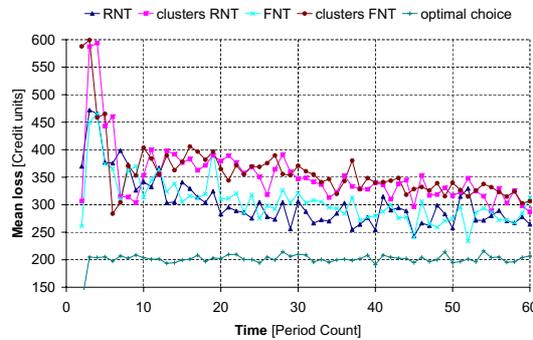

Fig. 5. Scenario with trustfulness independent of situation – all methods perform comparably

In the second scenario (Fig. 6), we have introduced a strong, but one dimensional situation dependence with one best provider per cargo type. We can see that in this case, context-based methods easily outperform the general trust and reach the optimum relatively fast. In Fig. 7, we can see that the depicted sub-market (defined by the contracts in one part of the context space) is rapidly dominated by the most trustful provider, while the others are restrained to the services where they perform better.
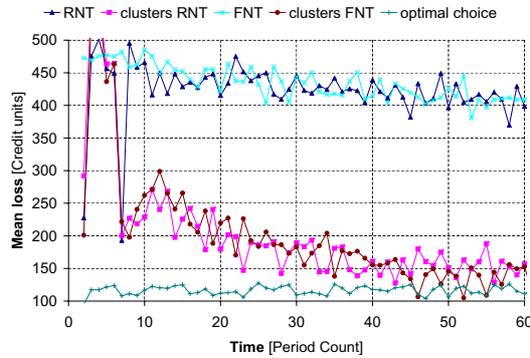
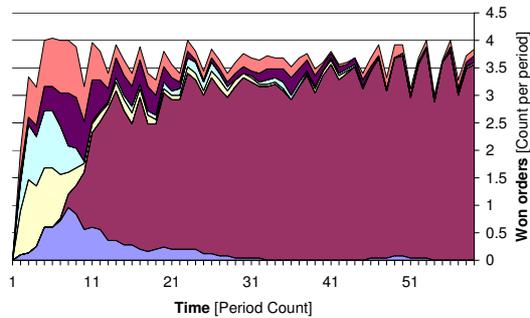**Fig. 6.** Scenario with trustfulness dependent on the cargo type only

Fig. 7. Market shares example with trustfulness dependent on the cargo type only
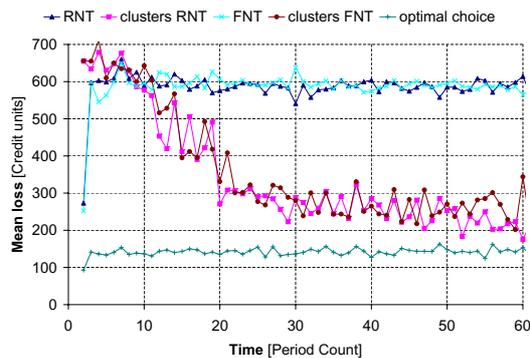

Fig. 8. Scenario with trustfulness dependent on all 3 parameters, with an adequate metrics

When we introduce a full 3D context dependence, we obtain the results shown in Fig. 8 and Fig. 9. We can see that the task is more difficult due to the increased dimensionality, but the context modeling solves the problem. The slower learning pace is clear when we compare the Fig. 9 with Fig. 7 – the market domination is slower.
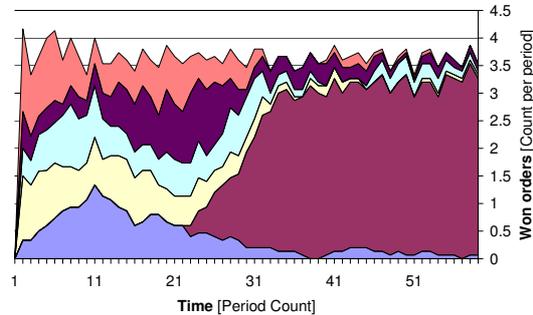
Fig. 9. Market shares with trustfulness dependent on all 3 parameters. Compare with Fig. 7.

5   Related Work

Besides the approach presented in this paper, there are many alternative approaches that attempt to relax the assumptions of the baseline trust mechanism as mentioned in the introduction. The FIRE model [9] presents two relevant techniques – role-based trust and certified reputation – that improve the reasoning about evaluated partners lacking interaction history. *Role-based trust* evaluates the partners on the basis of their social role, taking into account the endorsements of the others, guarantees or group membership. Evaluation of this trust component is performed using a set of rules applied on belief base of the trusting agent. In REGRET, the neighborhood reputation [8,6] and system reputation are proposed to achieve similar results, implemented by means of rules or aggregated reputation of the members of the evaluated agent's group.

Certified reputation, also proposed in [9], uses the ratings provided by the evaluated agent to assess its trustfulness. These ratings are created by the agents that interacted with the evaluated agent in the past, and are typically electronically signed. While this mechanism is valuable it is inherently biased – evaluated agent is free to chose only the ratings that are positive regarding its performance. Moreover, the electronic signature by itself proves only the *integrity* of the message. *Authenticity* of the message can be proved only if the evaluating agent can attach the signing key to a respectable (or at least known) information source. Otherwise, nothing can prevent agents from collusion, or creation of false evaluators and boosting of their ratings. Therefore, this technique is of limited interest if the identity of the agents is not proven.

The problem of the identity is also prominent in [15], where the agents (robots) are identified by their group membership, instead of the individual identity.

6   Conclusions and Future Work

In this paper, we have presented a mechanism that supports the use of existing trust models in the open, unmanaged and dynamic environments, where the agents can enter and leave the system at their own will, can change or disguise their identity easily and where the conditions under which the system operates are frequently changing.

We address the above particularities of the open environments by introduction of an intelligent mechanism that decouples the trustfulness from the statement of identity (e.g AID, address or name) and attaches the trustfulness to the important points in the Identity-Context metric space. It shall be noted that while their implementation is joint, the identity management and context representation parts of the module can be effectively split, using only half of the mechanism functionality. This is applicable fr example in the supply chain solutions, where the identity of the actors is certain and verifiable, but we still need to manage the context.

Besides the above-mentioned problem of securing open multiagent systems, trust models extended with the presented mechanism can be used in a completely novel contexts. In our current research [14], we experiment with use of such models for network intrusion detection and protection, in attempt to automatize the response to worm-type attacks. In this deployment, we don't evaluate individual hosts or applications, but rather the actual connections and their characteristics, taking into account the relevant statistics off the network traffic [16] as a context for the decision. The trustfulness of the connections is deduced from the status of the protected hosts and other alarms.

Such deployment changes completely the place of the trust model in the agent architecture – instead of being an accessory, protective module, it forms a core part of the business reasoning of the agent, under the assumption that rapid, automatic and intelligent response to attacks delivered by distributed security system is the best option how to address the worm threat. The underlying reasoning behind our approach is that the epidemics-like attacks in the virtual ecosystem shall be addressed by the approaches that are inspired by the real immune systems.

Acknowledgment

References

1. Marsh, S.: Formalising trust as a computational concept (1994)
2. Falcone, R., Castelfranchi, C.: Social trust: a cognitive approach, pp. 55–90 (2001)
3. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artif. Intell. Rev. 24, 33–60 (2005)
4. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multiagent systems. The Knowledge Engineering Review 19 (2004)
5. Stallings, W.: Cryptography and network security: principles and practice, 2nd edn. Prentice-Hall, Inc., Englewood Cliffs (1999)
6. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) Adaptive Agents and Multi-Agent Systems. LNCS (LNAI), vol. 2636, pp. 475–482. Springer, Heidelberg (2003)

7. Pechoucek, M., Rehak, M., Marik, V.: Expectations and deployment of agent technology in manufacturing and defence: case studies. In: Pechoucek, M., Steiner, D., Thompson, S.G. (eds.) AAMAS Industrial Applications, pp. 100–106. ACM Press, New York (2005)

8. Sabater, J., Sierra, C.: Social regret, a reputation model based on social relations. SIGecom Exch. 3, 44–56 (2002)

9. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. Journal of Autonomous Agents and Multi-Agent Systems 13, 119–154 (2006)

10. Rehák, M., Foltýn, L., Pěchoucek, M., Benda, P.: Trust model for open ubiquitous agent systems. In: Intelligent Agent Technology, 2005 IEEE/WIC/ACM International Conference. Number PR2416 in IEEE (2005)

11. Josang, A., Gray, E., Kinateder, M.: Simplification and analysis of transitive trust networks. Web Intelligence and Agent Systems 4, 139–162 (2006)

12. Yu, B., Singh, M.P.: Detecting deception in reputation management. In: AAMAS '03, pp. 73–80. ACM Press, New York (2003)

13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn., p. 654. John Wiley & Sons, New York (2001)

14. Rehák, M., Tožička, J, Pěchouček, M., prokopová, M., Foltýn, L.: Autonomous protection mechanism for joint networks in coalition operations. In: Knowledge Systems for Coalition Operations 2007, Proceedings of KIMAS'07 (2007)

15. Birk, A.: Boosting cooperation by evolving trust. Applied Artificial Intelligence 14, 769–784 (2000)

16. Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P.N., Kumar, V., Srivastava, J., Dokas, P.: Minds - minnesota intrusion detection system. In: Next Generation Data Mining, MIT Press, Cambridge (2004)