Formal Analysis of Models for the Dynamics of Trust based on Experiences

Catholijn M. Jonker and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
Email: {jonker, treur}@cs.vu.nl URL: http://www.cs.vu.nl/{~jonker,~treur}

Abstract. The aim of this paper is to analyse and formalise the dynamics of trust in the light of experiences. A formal framework is introduced for the analysis and specification of models for trust evolution and trust update. Different properties of these models are formally defined.

1 Introduction

Trust is the attitude an agent has with respect to the dependability/capabilities of some other agent (maybe itself) or with respect to the turn of events. The agent might for example trust that the statements made by another agent are true. The agent might trust the commitment of another agent with respect to a certain (joint) goal. The agent might trust that another agent is capable of performing certain tasks. The agent might trust itself to be able to perform some tasks. The agent might trust that the current state of affairs will lead to a state of affairs that is agreeable to its own intentions, goals, commitments, or desires.

In [1], [2] the importance of the notion trust is shown for agents, multi-agent systems, and their foundations. From the viewpoint of the users of agent systems Ousterhout [10] makes clear that work can only be delegated to such systems if they can be trusted without there being a constant need for inspection of their work. Elofson [4] states that the reach and effect of trust in the affairs of individuals and organizations is largely pervasive. Elofson continues with the problem that trust is somewhat illusive, difficult to define, difficult to create, and difficult to measure. Before focusing on the difficulties regarding the creation and measurement of trust, a brief survey is made of definitions of trust, for more information see [4], [5].

Trust of an agent in another agent (social trust) is sometimes defined as a kind of binary property, for example, an agent A trusting another agent B means that A believes that B will act in a way that is favorable to A, even though that act might not be most convenient to B at that moment [5]. A shorter variant is that of Demolombe [3]: “We can understand trust as an attitude of an agent who believes that another agent has a given property.” Another definition of trust, describes the notion as a subjective probability [5]. Common in these definitions is that the trusting agent A has a specific interest in the actions of the agent B that is trusted by A, and that B will act with respect to this interest even though it might seem that doing so is not favorable with respect to B's own interests. In [1], [2] this paradox is solved by the following definition of trust: "Trust is a theory and an expectation about the kind of motivations the agent is endowed with, and about which will be the prevailing motivations in case of conflict." This implies that an agent can have interests on several levels like economic interests, emotional and social interests (love, friendship, norms). They state that the mental ingredients of social trust are relative to the competence of the other agent, to the predictability of the behaviour of the other agent, and on the agents own faithfulness.

In the above definitions trust finally depends upon some sort of beliefs, predictions, or expectations. However, it is not clear (not meant as a criticism) where these beliefs and expectations come from. The definition of Lewis and Weigert [6] does not refer to beliefs or expectations, but to observations which in turn lead to expectations: "observations that indicate that members of a system act according to and are secure in the expected futures constituted by the presence of each other for their symbolic representations." Elofson [4] agrees that observations are important for trust, and he defines trust as: "trust is the outcome of observations leading to the belief that the actions of another may be relied upon, without explicit guarantee, to achieve a goal in a risky situation." Elofson notes that trust can be developed over time as the outcome of a series of confirming observations. From his experimental work, Elofson concludes that information regarding the reasoning process of an agent, more than the actual conclusions of that agent affect the trust in the conclusions of that agent.

The evolution of trust over time, also called the dynamics of trust, as mentioned by Elofson, is also addressed in [2]: "there is a circular relation, and more precisely a positive feedback, between trust in reciprocal delegation-adoption relations (from commerce to friendship)." An implication of this is that if an agent A trusts an agent B, then communicating his trust in B to B, can lead to an increase of B's trust in A. Of course, a similar feedback relation exists for distrust.

In this paper we consider trust from the perspective of the software agent, that is, trust within software agents regarding the reliability of objects and tools, their own work, the behaviour of others, and in the evolution of their environment (events and effects of actions performed by the agent).

Trust is based on a number of factors, an important one being the agent's own experiences with the subject of trust; e.g., another agent. Each event that can influence the degree of trust is interpreted by the agent to be either a *trust-negative experience* or a *trust-positive experience*. If the event is interpreted to be a trust-negative experience the agent will loose his trust to some degree, if it is interpreted to be trust-positive, the agent will gain trust to some degree. The degree to which the trust is changed depends on the trust model used by the agent. This implies that the trusting agent performs a form of continual verification and validation of the subject of trust over time. For example, you can trust a car, based on a multitude of experiences with that specific car, and with other cars in general. For this paper a formal analysis of the dependency of trust on experiences will be the central focus.

One of the key issues for the design of intelligent software agents is how trust is represented within the agent, and how the effect of experiences is specified. Representations can be *qualitative*, using specific qualitative labels (or term structures), or *quantitative*, using numbers as a representation. For example, trust could be measured by a real number between -1 and 1.

For a first analysis, a simple qualitative model is discussed in Section 2. In Section 3 the formal notion of trust evolution function is introduced, and properties of trust evolution functions are defined. In Section 4 trust update functions are introduced, and some properties are defined. Section 5 introduces a quantitative example model which takes into account an inflation rate on experiences.

2 A Simple Qualitative Model for Trust Update

In this section a simple qualitative trust model is discussed. The main purpose of this example is to identify a number of issues for further analysis.

2.1 The Representation of Trust

In the model considered in this section four trust values are distinguished and ordered in the following way:

unconditional distrust < conditional distrust < conditional trust < unconditional trust

The minimal trust value is unconditional distrust , the maximal value is unconditional trust. So, a first assumption on trust models is that there exists a set of trust values and they are partially ordered and maximal and minimal trust values exist.

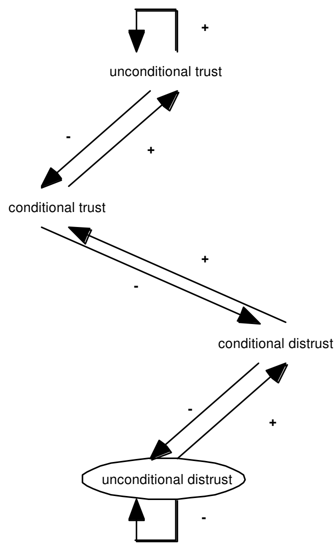

Fig. 1. A simple qualitative model for trust dynamics

2.3 Trust Characteristics

As discussed above, agents can have their own characteristics with respect to trust dynamics. There are many possible trust types of agents. To define these types two aspects can be taken into consideration:

(1) initial trust
(2) trust dynamics

Initial trust. With respect to initial trust the following possibilities can be distinguished:

1. *initially trusting*
   a) without previous trust influencing experiences the agent has unconditional trust: maximal trust value
   b) without previous trust influencing experiences the agent has conditional trust: a positive trust value, below the maximal trust value
2. *initially distrusting*
   a) without previous trust influencing experiences the agent has unconditional distrust: maximal distrust value
   b) without previous trust influencing experiences the agent has conditional distrust: a negative trust value, above the minimal trust value

Note that the actual trust values used by the agent can differ from the ones mentioned above. However, the trust values used will be partially ordered, and have maximal and minimal values.

Types of trust dynamics. The following trust dynamics types can be distinguished:

1. *blindly positive*:
   a) always unconditional trust
   b) definitive having trust: after a certain number, or sequence of positive trust experiences, the agent reaches the state of unconditional trust and will remain in this state indefinitely.
2. *blindly negative*:
   a) always unconditional distrust
   b) definitive losing trust: after a certain number, or sequence of negative trust experiences, the agent reaches the state of unconditional distrust and will remain in this state indefinitely.
3. *slow positive, fast negative* dynamics: it takes a lot of trust-positive experiences to gain trust, it takes only a few trust-negative experiences to lose trust.
4. *balanced slow*: slow dynamics both in positive and in negative sense
5. *balanced fast*: fast dynamics in positive and in negative sense. The *tit-for-tat* strategy is an example of a strategy that can be used for a balanced fast trust-type

6. *slow negative, fast positive* dynamics: it takes a lot of trust-negative experiences to lose trust, it takes only a few trust-positive experiences to gain trust.

Within the example used as an illustration in this section, the trust representation is just sufficiently rich to specify a difference in characteristics between slow and fast dynamics, but it is not rich enough to specify more subtle differences in characteristics. For example, it is not possible to specify that in an unconditional positive trust state, after three positive experiences trust will be blindly positive.

3 Trust Evolution Functions

In this paper trust is considered a mental agent concept that depends on experiences (evaluated events). One way to formally model the dynamics of trust is to formalise the dependency of trust on past experiences by a mathematical function that relates sequences of experiences to trust representations: a *trust evolution function*. Another way to formally model the dynamics of trust is in an inductive manner by a mathematical function relating a current trust representation and a current experience to the next trust representation: a *trust update function*. A natural question is whether these formalisations can be represented in terms of each other. In principle, any trust update function generates for any initial trust value a trust evolution function, but not every trust evolution function can be represented as a trust update function. Both ways of formalisation and their relations will be analysed in more depth in this (trust evolution functions) and the next sections (trust update functions and relations).

To obtain a formal framework, the following sets are introduced:

E A partially ordered set of experience classes
Examples are:
- $E = \{ -, + \}$ with $- < +$, as in Section 2, or
- an interval in the real numbers (e.g., $[-1, 1]$), or
- more dimensional variants.

Actually these representations denote evaluated events; for shortness the word experiences will be used.
In addition, E may have one or both of the following structures:
- two sets E_{pos} and E_{neg} indicating positive and negative elements of E, with $ev1$ negative and $ev2$ positive implies $ev1 < ev2$.
- a neutral element 0_E of E, such that
  $E_{neg} = \{ev \in E \mid ev < 0_E\}$ and $E_{pos} = \{ev \in E \mid ev > 0_E\}$

$\mathbb{N}$ The set of natural numbers.

ES The set $E^{\mathbb{N}}$ of experience sequences $e = (e_i)_{i \in \mathbb{N}}$ with $e_i \in E$; this set ES is partially ordered by:
$$\forall e,f \in ES: \quad e \leq f \Leftrightarrow \forall i \;\; e_i \leq f_i$$
For $e \in ES$ and $k \in \mathbb{N}$ by $e|k$ the finite sequence $(e_i)_{i \in k}$ is denoted

T A partially ordered set of trust qualifications
Examples are
- the set of trust qualifications in the example in Section 2, or
- an interval in the real numbers (e.g., $[-1, 1]$), or
- more dimensional variants.

In addition, T may have one or both of the following structures:
- two sets T_{pos} and T_{neg} indicating positive and negative elements of T, with $tv1$ negative and $tv2$ positive implies $tv1 < tv2$.
- a neutral element 0_T of T, such that

$$T_{neg} = \{tv \in T \mid tv < 0_T\} \text{ and } T_{pos} = \{tv \in T \mid tv > 0_T\}$$

Using these sets, the notion of trust evolution function can be formally defined; see Definition 3.1.

Definition 3.1 (Trust Evolution Function)
(a) A *trust trace* is a sequence

$$tt : \mathbb{N} \rightarrow T$$

(b) A *trust evolution function* is a function

$$te : ES \times \mathbb{N} \rightarrow T$$

Let $e \in ES$ and $i \in \mathbb{N}$, then $te(e,i)$ denotes the trust after experiences $e_0,\ldots,e_{i-1}$. Associated with every trust evolution function te, there is a function

$$te' : ES \rightarrow (\mathbb{N} \rightarrow T)$$

defined by:

$$te'(e) = (te(e,i))_{i \in \mathbb{N}} \text{ and } te'(e)(i) = te(e,i)$$

I.e., $te'(e)$ is a trust trace for every experience sequence e. Sometimes te is used to refer to te' as well.
(c) Trust traces and trust evolution functions are *ordered* by:

$tt1 \leq tt2$	iff	$tt1(i) \leq tt2(i)$ for all i
$te1 \leq te2$	iff	$te1(e) \leq te2(e)$ for all e

In Definition 3.2 a number of possible properties of trust evolution functions are formally defined. In this definition, *future independence* (see 1.), expresses that trust only depends on past experiences, not on future experiences. This is a quite natural assumption that is assumed to hold for all trust evolution functions; in particular, it holds for the example in Section 2. Also *monotonicity* (see 2.) is a quite natural assumption. It expresses that if the experiences are at least as positive (compared to a given sequence of experiences), also trust will be at least as positive (compared to the trust related to the given sequence of experiences). The example model discussed in Section 2 satisfies monotonicity.

The property *indistinguishable past* expresses that only the experiences themselves count and not the point in time at which they were experienced; in fact this property abstracts from the temporal aspect. We consider this not a natural property. Only in very simple cases it might be relevant, for example a trust evolution function in which just the number of all positive and negative experiences are counted and compared has this property; for example:

$$te(e, i) = [\#\{ i \mid e_i \geq 0 \} - \#\{ i \mid e_i \leq 0 \}] / i$$

Since this property expresses that experiences far back in time count just as strong as very recent experiences, all trust evolution functions that take into account some notion of inflation or forgetting of experiences will not satisfy this property (see, for example, Section 5). Also the simple example in Section 2 does not satisfy indistinguishable past. For example, a sequence of experiences

+ + + − + − + − +

leads to the value unconditional trust, whereas the sequence

+ + + + + + − − −

leads to unconditional distrust.

The properties *maximal* or *minimal initial trust* (see 4. and 5. in Definition 3.2) express the starting point of the trust evolution process. The properties of *positive* (or *negative) trust extension* (see 6. and 7.) express that after a positive (or negative) experience, trust will become at least as much (or as less) as it was. The example in Section 2 satisfies these properties.

The property *degree of memory based on window* n *back* expresses that only the last n experiences are relevant. All earlier experiences are forgotten. The example of Section 2 does not satisfy this property, not for any n. For example, the two experience sequences of arbitrary length

+ + + − + − + − + ... − +

and

− − − − + − + − + ... − +

will always lead to different trust values, even while the last part is equal. However, for not too sophisticated models for trust dynamics, this property might be relevant. It provides an easy way to specify the evolution, just by looking at the most recent experiences; e.g., the tit-for-tat strategy.

The property *degree of trust dropping* (or gaining) (see 9. and 10.) expresses after how many positive (or negative) experiences trust will be positive (or negative). The example of Section 2 satisfies degree of trust gaining and dropping 2: always after two positive experiences, trust will be positive, and always after two negative experiences, trust will be negative.

Four properties (see 11. to 14.) concern *limit behaviour*. They express, for different cases, conditions under which trust will become maximal (respectively, minimal). Essentially they express that it is always possible to reach maximal trust, if a sufficiently long period with only positive experiences is encountered, and the same for the negative case. The example in Section 2 satisfies the properties 12. and 14.; just take $N = M + 3$. Models for trust dynamics in which it is possible that a form of *fixation* occurs, i.e., so much of distrust is acquired that trust will not be possible anymore, independent of further experiences, do not satisfy these properties (see also the blindly positive or negative characteristics in Section 2.3.2). Properties 15. and 16. express this phenomenon of trust fixation.

Definition 3.2 (Properties of Trust Evolution Functions)

The following properties (in which $e,f \in ES$, $i,j,k,n \in \mathbb{N}$) are defined

1. *future independence*

a trust evolution function te is *future independent* if its values only depend on the experiences in the past:

if $e|k = f|k$ then $te(e, k) = te(f, k)$

2. *monotonicity*

   $e \leq f \Rightarrow te(e) \leq te(f)$

3. *indistinguishable past*

   if $e|k$ is a (temporal) permutation of $f|k$ then $te(e, k) = te(f, k)$

4. *maximal initial trust*

   $te(e,0)$ is maximal in T

5. *minimal initial trust*

   $te(e,0)$ is minimal in T

6. *positive trust extension*

   $\forall\ i,j\ \ [\forall k \in \mathbb{N} : i \leq k < j : e_k \text{ positive}\ ] \Rightarrow te(e,i) \leq te(e,j)$.

7. *negative trust extension*

   $\forall\ i,j\ \ [\forall k \in \mathbb{N} : i \leq k < j : e_k \text{ negative}\ ] \Rightarrow te(e,i) \geq te(e,j)$.

8. *degree of memory based on window n back* (forgetting about the past)

   $\forall\ i\ \ [\forall k \in \mathbb{N} : i\text{-}n < k \leq i : e_k = f_k\ ] \Rightarrow te(e,i) = te(f,i)$

   extreme cases:

   a) $n = 1$: only last experience counts

   b) $n = 0$: no experience counts

9. *degree of trust dropping* n

   $\forall\ i\ \ [\forall k \in \mathbb{N} : i\text{-}n < k \leq i : e_k \text{ negative}\ ] \Rightarrow te(e,i) \text{ negative}$

   extreme cases:

   a) $n = 1$: trust drops after 1 bad experience

   b) $n = 0$: trust is never given

10. *degree of trust gaining* n

    $\forall\ i\ \ [\forall k \in \mathbb{N} : i\text{-}n < k \leq i : e_k \text{ positive}] \Rightarrow te(e,i) \text{ positive}$

    extreme cases:

    a) $n = 1$: trust is given after 1 good experience

    b) $n = 0$: trust is always given

11. *positive limit approximation (continuous metric case)*

    if there exists an M such that for all $m > M$ it holds e_m is maximal, then for all $\varepsilon > 0$ there exists an N such that $te(e, n)$ is within at most ε from maximal for all $n > N$.

12. *positive limit approximation (discrete case)*

    if there exists an M such that for all $m > M$ it holds e_m is maximal, then an N exists such that $te(e, n)$ is maximal for all $n > N$.

13. *negative limit approximation (continuous metric case)*

    if there exists an M such that for all $m > M$ it holds e_m is minimal, then for all $\varepsilon > 0$ there exists an N such that $te(e, n)$ is within at most ε from minimal for all $n > N$.

14. *negative limit approximation (discrete case)*

    if there exists an M such that for all $m > M$ it holds e_m is minimal, then an N exists such that $te(e, n)$ is minimal for all $n > N$.

15. *negative trust fixation of degree* n

    if for some i the trust value $te(e, k)$ is minimal for all k with $i \leq k < i + n$, then $te(e, k)$ is minimal for all $k \geq i$.

16. *positive trust fixation of degree* n
if for some i the trust value te(e, k) is maximal for all k with $i \leq k < i + n$, then te(e, k) is maximal for all $k \geq i$.

4 Trust Update Functions

From a mentalistic perspective, the notion of trust evolution function suggests that an agent builds a representation for sequences of past experiences, and at each moment in time uses these representations of experiences to determine its trust. Another, from a computational perspective maybe more desirable model is that an agent does not build a representation of the (past) experiences, but only of trust itself, and that a new experience instantaneously leads to an update of the trust representation, without maintaining the experience itself. This perspective was also the perspective used in Section 2, and is addressed in more depth below. First the definition of a trust update function:

Definition 4.1 (Trust Update Function)
A *trust update function* is a function $tu : E \times T \rightarrow T$.

Note that Fig. 1 depicts an example specification of a trust update function. For a given trust update function, any initial trust value it generates by induction a unique trust evolution function te with te(e,0) = it. This relation between trust update functions and trust evolution functions will be addressed in more depth in Section 6.

Definition 4.2 (Properties of trust update functions)
The following properties are defined:
1. *monotonicity*
   ev1 ≤ ev2 & tv1 ≤ tv2 ⇒ tu(ev1, tv1) ≤ tu(ev2, tv2)
2. *positive trust extension*
   ev positive ⇒ tu(ev, tv) ≥ tv
3. *negative trust extension*
   ev negative ⇒ tu(ev, tv) ≤ tv
4. *strict positive monotonic progression*
   ev positive and tv not maximal ⇒ tu(ev, tv) > tv
5. *strict negative monotonic progression*
   ev negative and tv not minimal ⇒ tu(ev, tv) < tv

Note that all properties defined in Definition 4.2 hold for the example in Section 2. From a trust update function, by iteration for each initial trust value a trust evolution function can be generated. The following definition shows how.

Definition 4.3 (Trust evolution generated by a trust update function)
Let tu be a trust update function and it any (initial) trust value. The trust evolution function te *generated by* tu for initial value it is the trust evolution function te inductively defined by:

te(e,0) = it for all $e \in ES$

$$te(e, i+1) = tu(e_i, te(e, i)) \quad \text{for all } e \in ES,\ i \in \mathbb{N}$$

This generated trust evolution function is denoted by $te_{tu,it}$.

Properties of $te_{tu,it}$ relate to properties of tu, for example, in the following sense:

Proposition 4.4
Let tu be a trust update function. Then the following hold:

1. $te_{tu,it}$ is future independent
2. If tu is monotonic, then $te_{tu,it}$ is monotonic
3. If tu satisfies positive trust extension, then $te_{tu,it}$ satisfies positive trust extension
4. If tu satisfies negative trust extension, then $te_{tu,it}$ satisfies negative trust extension
5. If tu has strict positive monotonic progression and T is finite, then $te_{tu,it}$ has positive limit approximation
6. If tu has strict negative monotonic progression and T is finite, then $te_{tu,it}$ has negative limit approximation

Note the condition on finiteness of the set of trust values in 5. and 6. in Proposition 4.4. If T is infinite, then the condition of strict monotonic progression is not strong enough. For example, it might well be the case that the progression decreases to such an extent that it stays under a bound tv less than the maximal value. However, for the continuous case stronger notions of progression can be defined that guarantee that the maximal value is reached, for example: there exists a $\partial > 0$ such that $tu(ev, tv) - tv > \partial (maxtv - tv)$.

5 A Quantitative Example

The model for trust dynamics introduced in this section has as a basic assumption that there is some rate of inflation of experiences. Experiences further back in the past count only for a fraction of the recent experiences. For both E and T the closed interval [-1, 1] is taken. We assume there is an inflation rate of d (between 0 and 1; for example 0.5) per experience step. The following trust update function is defined:

$$g_d(ev, tv) = d\ tv + (1 - d)\ ev$$

In this trust function, after each new experience the existing trust value is multiplied by d (this expresses the inflation), and the impact of the new experience is added, normalised in such a manner that a 2-ary function from the interval [-1, 1] to [-1, 1] results.

(a) For a fully positive experience with value 1, the comparison with maximal trust value 1 is:

$$1 - g_d(1, tv) = 1 - [\, d\, tv + (1 - d)] = d\,(1 - tv)$$

This means that the distance of the trust value to the maximal trust value 1 is decreased to a fraction d of the old distance.

(b) For a fully negative experience with value -1, the comparison with maximal distrust value -1 is:

$$1 + g_d(-1, tv) = 1 + [\, d\, tv + (1 - d)(-1)] = d\,(1 + tv)$$

This means that the distance of the trust value to minimal trust -1 is decreased to a fraction d of the old distance.

(c) For a zero-experience, the following can be found:

$$g_d(0, tv) = d\, tv$$

This means that for a zero-experience the distance of the trust value to 0 is decreased to a fraction d of the old trust value.

The example trust update function defined in this section has the following properties: monotonicity, positive and negative trust extension, strict positive and negative progression. The trust evolution function generated by the example trust update function can be determined in an explicit formula as a sum of powers of d as follows:

$$f_d(e, k) = it\, d^k + (1-d)\, \Sigma_{i=0}^{k-1}\, e_{k-1-i}\, d^i$$

7 Discussion

In this paper a framework is presented that supports formal analysis of the dynamics of trust based on experiences. The formal models made within this framework can also be used for the specification of trust evolution and trust update for software agents as part of their design. The requirements imposed on models for trust dynamics can highly depend on the individual characteristics of agents, therefore, a variety of models that capture these characteristics is needed. The formal framework enables the explication of these characteristics. Both qualitative and quantitative example models are given that are based on explicit trust evolution functions and trust update functions with which these characteristics can be formally specified.

Trust may be influenced by experiences of different types. This paper models differences between experiences by mapping them into one overall set of distinct experience ‘values’. In addition, more explicit distinctions between different dimensions of experience could be made. Also other cognitive or emotional factors could be integrated. The work presented in [7], [8], [9] addresses some of these other aspects of trust, which could be integrated. This is left for future work.

References

1. Castelfranchi, C., and Falcone, R., Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification. In: Demazeau, Y. (ed.), Proceedings of the Third International Conference on Multi-Agent Systems, IEEE Computer Society, Los Alamitos, 1998, pp. 72-79.
2. Castelfranchi, C., and Falcone, R., Social Trust: Cognitive Anatomy, Social Importance, Quantification, and Dynamics. In: Proceedings of the First International Workshop on Trust, 1998, pp. 35-49.
3. Demolombe, R., To trust information sources: a proposal for a modal logical framework, In: Proceedings of the First International Workshop on Trust, 1998, pp. 9-19.
4. Elofson, G., Developing Trust with Intelligent Agents: An Exploratory Study, In: Proceedings of the First International Workshop on Trust, 1998, pp. 125-139.
5. Gambetta, D., Trust. Basil Blackwell, Oxford, 1990.
6. Lewis, D., and Weigert, A., Social Atomism, Holism, and Trust. In: Sociological Quarterly, 1985, pp. 455-471.
7. Marsh, S., Trust and Reliance in Multi-Agent Systems: a Preliminary Report. In: Proc. of the Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'92, Rome, 1992.
8. Marsh, S., Formalising Trust as a Computational Concept. Ph.D. Thesis. Department of Mathematics and Computer Science, University of Stirling, 1994
9. Marsh, S., Trust in Distributed Artificial Intelligence. In: C. Castelfranchi, E. Werner (eds.), Artificial Social Systems. Lecture Notes in AI, vol. 830, Springer Verlag, 1994, pp. 94-112
10. Ousterhout, J., Virtual Roundtable. In: Internet Computing on-line Journal, July-August issue, 1997.