Agent Reasoning Based on Trust and Reputation

J. Samek, F. Zbořil, Brno University of Technology, Czech Republic

SNE Simulation Notes Europe SNE 20(3-4), 2010, 21-28, doi: 10.11128/sne.20.tn.09985

In the area of intelligent systems development some deterministic or nondeterministic decision algorithms and mechanisms should be used to enable agents to behave intelligently. We are trying to enhance agent reasoning and especially agent decision making with a usage of trust and reputation of particular intelligent elements (agents) as well as some social groups. There can be large agent societies, where collaboration between agents is the best way and sometime the only possibility to achieve non-trivial goals. Often it is very difficult to find *best counterparts* for collaboration. Our approach works with trust and reputation principles which are inspired from real-world societies and we try to shift them into artificial societies to make their interaction and cooperation more effective.

Introduction

Trust is very important aspect in our everyday interaction with people, groups and institutions in our society. We should have a trust in the surrounding environment, people and institutions as well. We are often rated and judged on the basis of our trustworthiness and this defines a different manner of the interactions in our social life. We behave more openly towards subjects on account of the strong confidence and trustworthy subjects can access different types of information which can be confidential. In the case of abuse of the information, the trust of the subject rapidly decrease and it is usually very hard to restore it again.

Recent researches shows [5, 6] that system based on trust and reputation have great potentiality, for example in the e-commerce and autonomous distributed computer systems. This can be seen for example on the leading auction server *eBay*, where the selection of seller (from the buyer point of view) is based also on his or her reputation. All participants in the system are treated on the bases of his or her reputation. Trustworthiness of a seller so as of a buyer is represented by some value, which is update by the eBay system and depends on cumulating positive and non-positive ratings from other sellers or buyers. This reputation system, from our point of view, can be considered as relatively simple and closely aimed system.

In more sophisticated systems [3], we must deal with trust as strictly *subjective* and *context specific* metric, because it is assessed from the unique perspective of the element which has to trust somebody or somewhat and our interest is limited only to those actions (context) of a trustee that have relevance to the trust value.

In our proposal, we need to take into account many specific problems which come with *trust based reasoning*.

This paper describes preliminary proposal core for an *agent reasoning framework* based on trust and reputation principles. We proposed how a trustworthy value will *create/receive*, *store* and *represent* and *use* to agent decision. Our framework does not create next multi-agent architecture.

We are trying to build new layout based on known and well formalized bases (such as BDI [11]). This layout allow to agents to use trustworthy value to be more effectively in decision making and interacting with other agents.

The remainder of this paper is organized as follows: in Chapter 1, we describe theoretical background of trust and reputation in different disciplines of the real word; his typical characteristics and issues which are need to be take into account when are used in such context. Description of the core of our framework proposal – agent reasoning – is in Chapter 2. We go from some bases terms and notations and describe defined formulas. Finally, Chapter 3 concludes our paper, discusses open issues and our future work.

1 Trust and reputation meaning

1.1 Trust

Trust as an explicit concept is not the one that has a mutually accepted definition. We have identified the existence of trust and reputation in many disciplines of human behavior, for example: economists, sociologists and computer science [1, 2, 8].

In different areas we have different definitions as well as several different definitions in one discipline.

For our purposes, we adopt some following definition, which is used in computer science for the computation model of trust and reputation rating systems: trust is a subjective expectation an agent has about another's future behavior based on history of their encounters [1]. For our model, trust is internal rating (value) of each agent towards other agents in the system. It is based on bias or on reputation. Trust is evaluated in time, when is needed to make an agent decision, it's not persistent value in agent belief base and may vary in time.

1.2 Reputation

Reputation is an agent's mental attitude toward other agents gained during previous experiences (even indirect) with such agents. Based on trust meaning description, reputation in our model is realized as set of values which are given from past agent interaction or received *recommendation*. Reputation is stored in agent *belief* base (knowledge database or something else) when agent finished some interaction and made necessary evaluation or when agent receives some recommendation from other agent(s) in environment.

Typically, it is difficult to gain reputation from interaction in the large scale multi-agent systems. The interaction generally runs in small agent groups, where agents are close by distances or by their purposes. In the case, that these agent groups (or just each single agent from group) want to communicate with each other's is good to use recommendations. To get the best possible recommendation, we need to ask most trustworthy entity (agent) as we can. Recommendation trustworthy value and also self-trustworthy of target agent mainly depend on recommendation entity. If we trust to this entity, recommendation will be more valuable for our purposes.

There are many approaches and mechanism to ensure trustworthy entities in system. We can use PKI [10] – certification authorities and root authorities as we know from security of information systems. Toward to our approaches, it is more applicable to use *web of trust* [9] between agents and groups.

It allows us to use system more distributive without central entities – possibly points of failure. This web of trust is also more closely to the real word principles and is suitable to the agent and multi-agent systems principles.

1.3 Recommendation

The reputation value usually depends on recommendations. In recommendation process always participate three agents: the querying agent a_q, answering agent a_r (recommender) and the target of recommendation agent a_t. In the recommendation case, agent get indirectly trust value from recommendation agent to target agent [3]. This given recommendation value can be accepted as the agent's trust value to the target agent at or serves just for updating of the trust value previously counted. This recounting trust value depends on many aspects, also mainly on how trustworthy a recommender agent is.

1.4 Context and individualization of trust

There are many aspects, which comes with reasoning based on trust and reputation. These aspects are need to be take into account and will be described in this subsection. The primary aspect which is closely connected with terms such as trust and reputation is *subjective reception* and *individualization*. In a real word, each of us trust in such degree to our friends. This trust degree is based on his outer behavior but also is mainly depend on our internal "metrics", which we using to measure his trustworthy. This metrics are strictly individual for each of us.

Typical example is human quality "prejudice" – without knowing about something man X, based on his visage (for example) we make opinion to his trustworthy. Someone, who also does not know X, makes another opinion, which can be absolutely different from our opinion. The same visage, the same man, the same knowledge about him may mean different trustworthy into him.

This is just simple example to demonstrate that trust is strictly subjective and mainly depends on our internal evaluating our perceptions for each entity (human, agent). This perception and internal evaluating may vary in time – it depends on ability of evaluating entity: *learning in time* based on previous *experiences*. In different cases, the perceptions may by for all entities the same (each agent have same sensors) but internal evaluating are different.

Perception is represented in to internal agent mental state and based on agent knowledge is differently interpreted – in this case, we call it as *agent personality*.

Another very important aspect is that trust and reputation are both *context dependent* [1, 7].

It means that trust and reputation are not one-dimensional values – they are at least two-dimensional.

We must say in which context the entity is trustworthy, if we talk about entity trust worthiness. We can't simply say: "he is trustworthy" or not. He or she must be trustworthy in some context – in some quality.

Context may be for example: "can cook" or "economic advice". If we need advice in some economic problem, we ask someone who is trust worthy in context "economic advice", because advice from someone who is trustworthy in "can cook" in our economic problem may not be fine. In the next case, one entity may be in some context trustworthy and in another not.

For example: if our friend Bob is a doctor, then hi is trustworthy in the context "can save our life", but if we need to cook apple pie, we will go for someone who is trustworthy in the context "can cook". So, Bob is trustworthy as a doctor, but he is untrustworthy as a chef.

With this context aspect many other problems and open issues come. At first, if we would like to evaluate some experiences after an interaction, we need to decide in which context or contexts the interaction was done. Based on this decision, we may update our belief base and finally we can do interaction evaluations.

From one interaction different reputation value in different contexts may be obtained. Another important but implementation difficult aspect is *reputation transference* – transference of one's reputation from one context to another [2]. For example: when we know that Bob is trustworthy as doctor, does it means that Bob is trustworthy as chef or not trustworthy as chef – is this decidable?

This problem may be decided on the bases of context similarity – we need to find algorithm which is able to compare two different context (context is composed from attributes – will be described in the section 3.3) and decide *similarity degree* between them.

Similarity degree allows us to decide if the transfer from one context to another is possible. This transference problem is quite complex problem and is outside the scope of this paper.

Two cases of trusts and reputation contexts in the system are possible [1]:

1. *Uniform context*. In the uniform context environment, we rate all the agents in the same context (every agent is related to the same subject matter). For example, we have a set of agents providing email service which have related attributes, so we can rate every agent in this service context. We omit all others context in this simple mail service system and we do not define context for reputation because it is known and only one.

2. *Multiple contexts*. In the second case, we have multiple context environments. In the multiple contexts environment, any agent's reputation is clearly context dependent. We need to take into account similarities and differences among the contexts. Transference of one's reputation from one context to another may be used.

In our framework proposal, we use multiple contexts environment, which is most suitable for distributive multi agent systems and reflect the real world principles.

2 Framework for agent reasoning

Before we start to formalize our framework core components, we need to show from which phases *the reputation is built and trust evaluating process* is composed. It allows us to understand following formal notation and the used principles.

2.1 Reputation building and trust evaluating

If we want to make decision based on trust value, we need to do some steps. First of all, we need to do some monitoring of trustee performance – *monitoring phase*.

Based on this, we make some experiences with trustee or we gather some information about him or her from the reputation. Asking for a reputation of trustee is used, when direct monitoring – experience of an agent is not possible.

From the phase of monitoring of an agent's performance we need to interpret some facts, store them into some belief base (knowledge base) and then we make decision if this experience was good, bad or neutral.

This phase is called *interpretation phase*. Recommendation process, when another agent (recommender) gives us some information about trustee is also kind of experience and they also need to be stored in agent's belief base.

The experiences in the belief base needs to be stored with *time stamps*. This means that every interaction or recommendation stored in the belief base will be dated with unique (actual) time stamp. This is useful to ensure that negative or positive experience gained long time ago will have not the same impact as *fresh experience*.

After the interpretation phase, the trust value *evaluation phase* can start. Given set of experiences in a time allow us to use trust update algorithm which update agent trust value in a context. This algorithm has many different inputs – such as agent mental states, agent individual preferences, environment specific preferences and so on. There is out of scope of this paper to describe trust evaluating process, this will be our task for future work.

From all the previous phases, final ensured trust value can be used as one of many input parameter for agent decision making. If the agent's decision will be evaluated as satisfying or not, agent can increase or decrease weight function based on trust value parameter in the future decision making process.

2.2 Trust and reputation value representing

In some models [2, 4] the trust/reputation value is represented as a binary value r, typically $r \in \{0,1\}$, it means $r \in \{\text{untrustworthy}, \text{trustworthy}\}$. In our framework, we would like to express such kind of partial trustworthy or partial untrustworthy for modeling trust and recommendations effects closely.

Toward this, we define trust value as natural number in an interval $r \in \langle x, y\rangle$, where x represent the worst possible rating and y represent the best possible rating of agent's trustworthy.

It is not important if $x = 0$ and $y = 100$ or $x = -100$, $y = 100$. Decisions about this interval will be implementation specific. However it is important to ensure that the trust value must change from x to y with difference Δr, which respect to model requirements and trust evaluating manners of the agent system.

2.3 Framework notation

Basis entity of each agent system is an agent. We define set of agent A as set of all possible agents in the system:

$$A = \{a_1, a_2, a_3, \dots, a_n\}.$$

To store reputation or incoming recommendation into the belief base and to make trust evaluation process it is need to determine context in which the reputation or recommendation was done.

Toward this, we need to define context. In our proposal the context definition is based on the terms *attribute* and *attribute domain*. Attribute domain means possibly range of attribute. So, we define set of all possible attribute domains Ω, when each element from this set is a domain:

$$\Omega = \{\omega_1, \omega_2, \omega_3, \dots, \omega_m\}.$$

One domain ω_1 may be for example set of *natural numbers* $\omega_1 = \aleph$, next domain ω_2 for example set of real numbers $\omega_2 = \Re$, boolean type $\omega_3 = \{0, 1\}$ or set of named constants (enumerated type) $\omega_m = \{\text{large, big, huge}\}$, etc. Finally, we define set of all attributes B, where each attribute from this set is always projected to such domain:

$$B = \{b_1, b_2, b_3, \dots, b_k\},$$
$$\forall b \in B\ \exists \omega \in \Omega\colon\ \mathrm{dom}(b) = \omega$$

Attribute b_1 may be for example Intelligence Quotient value (IQ). We can define domain ω_1 for this attribute as set of natural numbers in range from 0 to 200: $\omega_1 = \langle 0, 200\rangle$ a $\omega_1 \subset \aleph$. The tuple – attribute and his domain – can be written as $\langle \text{IQ}, \omega_1\rangle$.

Example of attribute sex (as an example of another attribute b_2) based on named constants domain: $\text{sex} \in \omega_2$, $\omega_2 \in \{\text{male, female}\}$.

At this moment, we can use previous definition to define the term *context*. We can theoretically define context as a set of tuples: *attribute* × *value*, where *value* is element from the *attribute* domain.

For example, context "intelligent male" or "intelligent female" may be defined as follows:

- "intelligent male" = $\{(\text{IQ}, 100), (\text{sex}, \text{male})\}$
- "intelligent female" = $\{(\text{IQ}, 100), (\text{sex}, \text{female})\}$

But in this context definition, there is problem to express some kind of *inequality*. In the previous example we can see that "intelligent male" is only the male who has exactly the same IQ as number 100. Actually, every male who have IQ equal or greater than 100 may be "intelligent male". Toward this, we need to add new element into context definition, this element will define range of values which attribute can take. This element represents an *operator* and we define set ‡ as set of all basic operators:

$$\Theta = \{=, >, <, \leq, \geq, <>\}$$

These operators have meaning of usual relation operators. Theirs application to the domains ω of some attribute creates range of values, which is a subset of ω. For each attribute domain $\omega \in \Omega$ it is necessary to define a function, which makes a mapping for each operator and some parameters to a subset of the original domain.

When the usual mathematical sets and the usual operators are used the evaluation is simple: in the domain ω_1 for attribute IQ from the previous example the result of application $(\mathrm{IQ}, \geq, 100)$ is the range: $(100, 200)$.

For other cases, where attribute domain is for example an enumerated type or other special domains, they should be evaluated by a function defined explicitly. Result of application $(\mathrm{sex}, >, \mathrm{female})$ is undefined without special function, which defines the result of these comparison. In the other cases application on the same domain is transparent: $(\mathrm{sex}, =, \mathrm{male})$ results $\{\mathrm{male}\}$; there is no need for a comparison function definition.

Finally, we can define context as set of triples: attribute × operator × value (from attribute domain); and set of all context C as follows:

$$C = \{c_1, c_2, c_3, \ldots, c_l\}$$
$$\forall c \in C : c \subset B \times \Theta \times \Omega_B$$

From all the previous definitions, we provide basic terms definitions toward our notation: *trust, reputation* and *recommendation*. Trust in our proposal is defined as a function T. Result of this function is actual trustworthy value from some unified domain $\omega \in \Omega$ (described in Section 2.2) in such context $c \in C$ into another agent $a \in A$.

$$T : A \times C \to \Omega$$

As we say in Section 2.1, we need to ensure that recommendation and reputation will be marked with some timestamp, which allow us to use more relevant information in the belief base. Timestamp help us to determine freshness of this information. At this point, we define time set TS as set of all time units in which interaction updates belief base was done.

$$TS = \{ts_1, ts_2, ts_3, \ldots, ts_x\}$$

In a *recommendation* function, we need to implement source of recommendation (recommender agent) and target of recommendation (target agent). As we say above, it is very useful to know when the recommendation was done. So we can define recommendation function E which maps agents, context and time moment to a value from an attribute domain.

$$E = A \times A \times C \times TS \to \Omega$$

And finally, *reputation* function R is defined as a mapping to a gained value on some unified domain from a target agent in such context in time – it is defined as follows:

$$R = A \times C \times TS \to \Omega$$

2.4 Agent belief base

We provided formal bases of our framework in the previous chapter. This chapter extends these bases and define how information is interpreted and stored in agent's belief base. This description is provided from the point of view of an evaluated agent. In our framework, we recognize three sources of information to evaluate trustworthy. These sources are:

1. *Recommendation* – information about an agent trustworthy in a context, this is obtained indirectly from another agent.
2. *Reputation* – information about an agent trustworthy in a context, this is obtained directly from own experience with her or him; or this is obtained from observing or premises.
3. *Facts* – information about an agent attributes – qualities.

Last mentioned sources are the facts. Facts about agents are created and updated in time and they are based on some received recommendations or they are based on reputation.

We define fact with a function k, where inputs are an agent $a \in A$ and attribute $b \in B$. Result of this function is a value from attribute b domain and an operator $\theta \in \Theta$.

$$k: A \times B \rightarrow \theta \times \Omega_B$$

For example the fact about agent a_1 (in respect to example from the previous chapter where attribute is IQ and his domain is in the range $\langle 0{,}200 \rangle$ write the following: $k(a_1, \text{IQ}) = (=, 100)$ – which means: we know, that agent a_1 has attribute (quality) IQ and this attribute is equal to the value 100 (from attribute domain $\langle 0{,}200 \rangle$).

Retrieving and maintaining the facts about other agents are needed for inferencing another attributes and for building another reputation in such context based on the inferred attributes. If we know that context c is composed from some set of attributes and we have no direct experience in the context c, we can build default trustworthy from the known attributes obtained from other contexts. This inferencing deals with *reputation transference* – described in Section 1.4.

At this moment we can provide simple example of attribute inference from some reputation:

- Let the context c "intelligent male" be defined as: $c = \{(\text{IQ}, >, 100), (\text{sex}, =, \text{male})\}$,
- reputation of agent a in a context c "intelligent male" is 100, which means (in a unified reputation domain) maximal trustworthy,
- we can infer from this reputation two facts:
  - $k(a, \text{IQ}) = (>, 100)$,
  - $k(a, \text{sex}) = (=, \text{male})$,
- let context c_2 "male" be defined as: $c_2 = \{(\text{sex}, =, \text{male})\}$,
- let context c_3 "intelligent" is defined: $c_3 = \{(\text{IQ}, >, 100)\}$,
- we can infer reputation from the facts for a in context c_2 and c_3 without direct experience or without given recommendation in these context:
  - $R(a, c_2, \text{time}_x) = 100$,
  - $R(a, c_3, \text{time}_x) = 100$.

This very simple example of inferencing and reputation transference shows, that it is possible to infer reputation from the facts, respectively infer facts from the reputation. In some complicated cases, similarity degree must be used to decide which attributes can be inferred and which cannot be inferred.

2.5 Trust evaluation

Based on definitions mentioned in the previous subsection, we propose in less formally way the trust evaluation algorithm. In this evaluation process we must combine reputation history with recommendations. Results of this evaluation are used for agent decision making about with whom it is good to cooperate and with whom it is not good.

After each interaction or received recommendation, the agents can make an evaluation and update their belief bases. On the bases of such evaluations the trust value of their counterparts is updated. Evaluation mainly depends on the reputations and facts. In a case when no interaction has been made in the past and no reputation value has been set, the agent uses some default politics to bind initial trust value into some "default value". There are many default politics to bind default trust value, for example:

- "paranoid" – the agent never trusts anyone until he or she prove his or her own trustworthy fairly,
- "neutral" – the agent takes a neutral position, it is capable to cooperate on the bases of positive recommendation,
- "friendly" – the agent is open to cooperate with anyone without previous experience.

This default politics may vary in time for each agent. In typically cases when an agent is new in an agent system, he is "friendly" and he is trying to make more friends. After time, when he was well profiled in the system and is trustworthy in his perimeter, it may change our politics to "neutral" or "paranoid" for example.

Building agent interaction history (reputations set) can be called to be *learning* process. Generally, agent increase trust to another agent, if he or she evaluates interaction as "satisfying" [7]. In "not satisfying" case, agent decrease the trust value.

During the agent learning process, if the decision of interaction (cooperate/defect) is based on other agent recommendations, the agent will also update its trust after any agent gives a recommendations.

For example, if *Alice* recommend to *Carol* that *Bob* is very good auto mechanic and *Carol* decide to go to *Bob* for her car repair, then *Carol* update trust into *Alice* also in such context as “recommendation” if will be satisfied (or not) with *Bob* service.

2.6 Agent decision based on trust

There are a lot of input parameters which can enter agent decision procedure, and the trust value can be one of them. In our agent system, we suppose that trust value is one of the main input parameter. We propose the *decision function*, which uses agent belief base – facts, reputation and recommendations – and maps it in a simple case to a binary value: cooperate/refuse (true/false, +/–). This value enters the decision procedure as a recommendation parameter to interact or not.

There are many variants of *decision functions* value types (ranges); they can be defined also as domain of attributes. For example, in a sophisticated case, the return value can be defined on interval $\langle -2,2 \rangle$, which may mean:

- -2: strong recommendation – do not interact,
- -1: light recommendation – you should not interact
- 0: no recommendation (unable to evaluate recommendation or neutral position),
- 1: light recommendation – you should interact,
- 2: strong recommendation – do interaction!

Internal evaluation mechanism of *decision functions* can be generally describes as follow. At first, agent must estimate some *threshold* value which is compared to trust domain range and defines delimiter for assignation return value.

If the internal trust value into agent in such context was higher or equal, agent decide to return “+”, otherwise “–” (depends on return value domain). For example, we estimate *threshold* to 80 and our *trust* to an agent is 90: the resulting value was then “+” (for a simple case) or “2” (for a sophisticated case).

Estimate function for threshold value differs due to agent metal state and many other aspects. To define threshold as a constant (for example 0.5) is a simple way to implement the estimate function. More sophisticated algorithm may use history of interactions: for example pair “*cooperate*” decision with “*non-satisfied*” results of interaction and update threshold value toward this. It is out of scope of this paper to define all implementing variants for estimate function.

2.7 After decision belief base update

If an agent decide to *interact* and it is based on *trust decision functions*, the feedback from interaction (agent was satisfied or not) update agent belief base. Agent updates our interaction history and may update trust to recommenders when interaction was made based on recommendation. We combine interaction history with feedback value to provide probability of next successful interaction in such context.

Updating reputations into each recommender after every interaction which was made on the recommendation based is also complex problem. We need to deal with feedback value, given reputation value and interaction history in the context “recommendation” for each of the recommenders.

This recommender rating is also very important for building set of agents, which are good in the recommendation context and which are not. This learning process allows us to be more effectively in time.

3 Conclusion and future work

In this paper we present preliminary framework proposal for multiple context model of trust and reputation which may allow agent reasoning based on trust. We describe critical common trust and reputation problems which are needed to be taken into account in solving reasoning problem based on trust principles. This proposal is based on known interaction protocols for the most used agent architectures such as BDI. Agents build their belief base: stores interactions history retrieves recommendations and infer facts and infers decisions.

Our model makes explicit difference between trust and reputation. We define reputation as a quantity inferred from interactions which can be highly relative toward to evaluating agent mental state and the interaction history.

We define trust as agents (trustor) internal quantity toward to trustee in a context. It can be inferred from facts or from reputation and recommendations about the trustee. It always represents strictly individual metrics. We show that trust and reputation ratings should be context and individual dependent quantities.

The framework notation, which was presented, allows us to simulate our proposal in future work. We will concentrate on formalization of the trust evaluating process before we simulate the system model. Also there are still a lot of works on formalization context transference process and context inference from agent attributes facts.

These tasks are very complex problems and must be well mapped to provide more effectively trust *decision function*, which is a core of our framework.

Acknowledgement

The research has been supported by the project MSM0021630528 and grants GP 102/07/P431 and GACR 102/09/H042.r

References

[1] L. Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD thesis, Massachusetts Institute of Technology, 2003.

[2] L. Mui, M. Mohtashemi, and A. Halberstadt. *A computation model of trust and reputation*. In Proc. 35th Annual Hawaii International Conference on System Sciences (HICSS'02), volume 7, page 188, 2002. Casti, J. L.: Reality Rules – Picturing the World in Mathematics: I, II. Wiley, New York, 1992.

[3] J. Samek and F. Zboril. *Multiple Context Model for Trust and Reputation Evaluating in Multi-Agent Systems*, In: Proc. of CSE 2008, Košice, SK, 2008, pages 336-343, ISBN 978-80-8086-092-9.

[4] S. Sen and N. Sajja. *Robustness of reputation-based trust: Boolean case*. In AAMAS '02: Proc. 1st Int. joint conference on Autonomous agents and multiagent systems, pages 288–293, 2002.

[5] P. Resnick and R. Zeckhauser. *Trust among strangers in internet transactions: Empirical analysis of Ebay's reputation system*. In NBER Workshop on Emperical Studies of Electronic Commerce, 2000.

[6] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. *Reputation systems*. Commun. ACM 43, 12 (Dec. 2000), 45-48

[7] Y. Wang and J. Vassileva. *Trust and reputation model in peer-to-peer networks*. In P2P '03: Proc. 3rd Int. Conf. on Peer-to-Peer Computing, page 150, 2003.

[8] A. Abdul-Rahman, S. Hailes. *Using recommendations for managing trust in distributed systems*. In: Proc. IEEE Malaysia Int'l Conf. on Communication 1997.

[9] Abdul-Rahman. *The PGP Trust Model*. EDI-Forum, April 1997.

[10] Wikipedia. *Public key infrastructure — Wikipedia, the free encyclopedia*, 2008. http://en.wikipedia.org/wiki/Public_Key_Infrastructure, accessed 24-11-2008.

[11] M. Wooldridge. *Reasoning About Rational Agents. The MIT Press*. 2000. ISBN 0-262-23213-8.

Corresponding Author: J. Samek,
Brno University of Technology,
Department of Intelligent Systems,
Božetěchova 2, 612 66 Brno, Czech Republic;
samejan@fit.vutbr.cz

Received & Accepted: MATHMOD 2009 -
Revised: September 10, 2009 -
Accepted: June 10, 2010 -