*Key words:*
*Trust, reputation, trust management, intelligent agents, agent systems, adaptive behavior, e-commerce*

Costin BADICA[♠], Maria GANZHA[*#♦], Maciej GAWINECKI[*♦], Paweł KOBZDEJ[*♦], Marcin PAPRZYCKI[*♣♦]

# TOWARDS TRUST MANAGEMENT IN AN AGENT-BASED E-COMMERCE SYSTEM – INITIAL CONSIDERATIONS

Trust plays an important role in any commercial system. It also has to be conceptualized for an e-commerce system in which all transactions are autonomously performed by software agents. In this paper we make the first attempt at specifying how trust will appear and be dealt with in our model system. In particular, we focus our attention on how the trust relationship between shops and buyers visiting them has been influenced by our specific e-commerce modeling and system design decisions.

## 1. INTRODUCTION

In our recent work we have developed and presented the first implemented version of an agent-based model e-commerce system [3, 4, 5]. In this system there exist a number of places where its behavior is influenced by what can be defined as a "trust relationship" between its components. For instance, we assume that buyers, who would like to participate in price negotiations may or may not be admitted depending on their past behavior. Specifically, if a buyer agent *X* representing buyer *mpaprzycki* visited e-store owned by *cbadica* and each time *X* participated in price negotiations it came out as a winner, but never completed a purchase, then the *cbadica* e-store may decide that representatives of *mpaprzycki* are no longer welcomed in the store as they are just acting as spoilers who prevent other legitimate buyers from completing their transactions (maybe they represent a competitor that decided to not to play fair).

In this context, the aim of this paper is to look in more detail into: (1) where in our

♠ University of Craiova, Craiova, Romania
* Systems Research Institute, Polish Academy of Science, Warsaw, Poland
\# Elbląg University of Humanities and Economics, Elbląg, Poland
♣ Warsaw School of Social Psychology, Warsaw, Poland
♦ Work partially sponsored by a Maria Curie IRG grant (project E-CAP)

system we have to deal with trust management, (2) which "events" that take place in the system influence trust relationships, and (3) how do they influence them. We proceed as follows: in the next section we briefly outline existing approaches to trust management and follow with a concise description of our system (focused on its trust-related features). In Section 4 we discuss events influencing trust relationships. Finally, in Section 5 we suggest how shop's trust in buyers can be represented.

## 2. APPROACHES TO TRUST MANAGEMENT

While there is no full agreement in the definitions of trust and reputation, in the literature these two terms are often understood in a way similar to: (1) ***trust*** – a peer's belief in another peer's capabilities, honesty and reliability based on its own direct experiences; (2) ***reputation*** **–** a peer's belief in another peer's capabilities, honesty and reliability based on recommendations received from other peers.

Thus, trust is typically conceptualized as a one-to-one relationship (I trust her based on my experiences in dealing with her), while reputation is a one-to-many relationship (I trust her because others also seem to trust her). Furthermore, both trust and reputation are based on long-term relationships and are cumulative (e.g. repeated positive actions result in building a strong relation of trust).

There exist a number of ways of computing reputation and trust (due to the lack of space we just list them and refer readers to presented references for further details): (1) summation or average of rankings [18], (2) Bayesian systems [21], (3) discrete trust model [1, 2, 9, 10], (4) belief models [13, 14, 22, 23], (5) fuzzy models [15], and (6) flow models [17]. Each one of them aims at providing a specific measure of trust and reputation that, for instance, can be used in constructing various rankings.

Currently, there exist two basic types of trust management systems: *centralized* and *decentralized* (see [1, 7, 10] for extended discussion of this topic). In *centralized reputation systems*, trust information is collected from community members (in the form of ratings). The central authority collects all ratings and derives a score for each peer (and posts it for others to see). A typical example would be rating system used in eBay, where both sellers and buyers can be rated by their counterparts, while the eBay system calculates updates and posts actual rankings for all eBay users to see.

In a *distributed reputation system* there is no central location for submitting ratings and obtaining resources' reputation scores. Instead, there are distributed stores where ratings can be submitted. Typical examples of such a situation are peer-to-peer systems. In a "pure" peer-to-peer setting, each user has its own repository of trust values of resources she knows. Individual trust-values are then shared by peers and utilized in a collaborative process to build an aggregated trust-value of a resource.

Let us now describe our proposed e-commerce system and return to the applicability of the above presented models of trust ad reputation within its settings.

## 3. SYSTEM DESIGN AND ITS EFFECT ON TRUST MANAGEMENT

The proposed agent-based e-commerce system models e-marketplace where *shop agents*, represent *User-Seller*s and attempt at selling products to *buyer agents* representing *User-Client*s. Use case diagram of the system is presented in Figure 1. Here, we focus our discussion on issues that are most important for trust management. A complete description of the system can be found in [3, 4, 5].

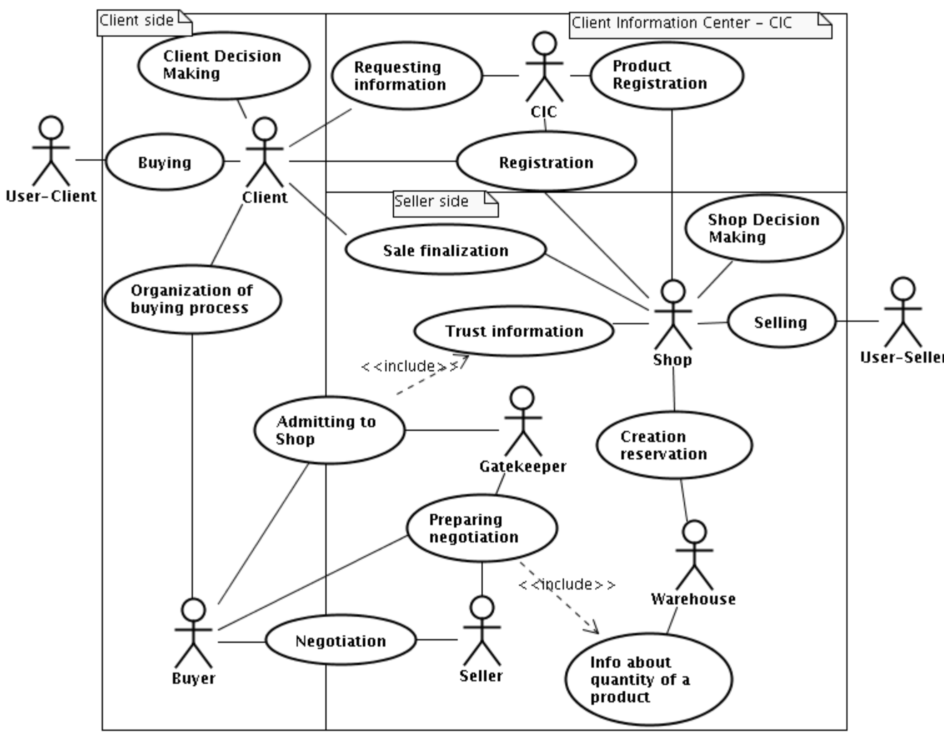


Figure 1. Agent-based e-commerce system – use case diagram

We can distinguish three major parts of the system: (1) the *Client Information Center* where white-page and yellow-page type data is stored – our current solution of the matchmaking problem, (2) the *Client side* where agents and activities representing *User-Client* take place, and (3) the *Seller side* where the same is depicted for the *User-Seller*. Let us now briefly describe agents represented in Figure 1.

The *CIC agent* (*CIC*) is responsible for providing information which e-store in the system sells which products. Information about products and stores is semantically represented – using OWL Lite demarcation and persisted in a Jena [12] environment (for more details see [11]).

Within the *Client side* we have the *Client agent* (*CA*) that represents its *User-Client* in autonomously making all necessary decisions related to the purchasing process and multiple *Buyer agents* (*BA*) which actually take part in price negotiations.

The *Seller side* consists of a number of agents that facilitate product sales. The *Shop agent* (*SA*) is the central manager of the e-store and autonomously makes all decisions pertinent to selling products offered by the store. The *SA* is helped by (1) the *Gatekeeper agent* (*GA*) that is responsible for admitting (or not) *BA*s to the host, management of the process of preparing negotiation which includes, among others, registration of participants and supplying them with negotiation template and protocol, and releasing *BA*s to price negotiations; (2) the *Warehouse agent* (*WA*) that is responsible for inventory and product reservations management; and, (3) multiple *Seller agents* (*SeA*) that are directly involved in price negotiations with *BA*s.

A typical usage scenario is as follows (for a detailed description see [3, 4, 5]). Let us assume that system is already initialized and all information about all products sold by e-stores has been registered with the *CIC*. *User-Client* formulates a request – what product she would like to purchase. The *CA* queries the *CIC* which stores sell thought after product and attempts to "deliver" a *BA* to these stores it deems worthy of its trust. Depending on the trust that particular stores have in the *CA*, its *BA*s are allowed to enter (or not), participate in price negotiations and report results to the *CA*. Based on obtained results, the *CA* makes decision to (1) attempt purchase at one of the stores, (2) to try negotiate a better price or (3) to abandon purchase altogether.

In our system we utilize an *airline ticket reservation mechanism* to manage the purchasing process. Successful price negotiations result in a reservation being issued to the winner. Within a certain time, specified in the reservation, that winner can make a purchase of the product at negotiated price. When the reservation expires the reserved product is returned to the pool of available products and the only way for the *BA* to make a purchase it through repeated participation in price negotiations. Let us now discuss in more detail issues related to trust management in the system.

## 4. TRUST RELATIONSHIPS AND CHANGES IN TRUST IN THE SYSTEM

Let us first return to the two notions introduced in Section 2. While it is quite possible to introduce into our system the concept of reputation (e.g. the *CA* can communicate with other *CA*s and ask what is their opinion about a particular *SA*), for the time being this type of agent behavior is not supported. Therefore, we will focus our attention on trust. The definition presented above makes it clear that trust can be viewed as a complex phenomenon. Let us, however, take into account the context of e-commerce and consider for a moment the notion of an *image.* Here, let us contemplate the following simple scenario. Store *A* is known to deliver goods on time, always correct goods, etc. But it is also known to overcharge for goods and services. Such a store can have a high ***positive*** value of *trust*, but have *image* a part of which (for some) may be rather ***negative*** (store that values itself too highly). Therefore, we would like to conceptualize *trust* as directly based on "hard facts" (facts that could be a part of a "service level agreement" and could be directly measurable). On the other hand *image* is going to be viewed as a broader concept that involves also individual perceptions / emotions (and as a result we would like to claim that $trust \subseteq image$).

### 4.1 CLIENT TRUST

Let us now consider trust as perceived from the point of view of the *User-Client*. Here, the *Client agent* (*CA*) represents interests of its *User-Client*, and its actions depend directly on its image of particular e-shops.

Currently we assume that price negotiations are fair (there are no illegal mechanisms used to raise the final price; or, when a given product is sold by multiple stores they do not collude with each other to raise prices, etc.). We also assume that agents "cannot" disguise their ownership (i.e. they can take up a new identity to wipe out bad trust-value (fresh start), but they cannot pretend to represent someone with a good trust-value). Thus, when considering interactions between the *CA* and an e-store, there are three moments that affect the image of the store "in the eyes" of the *CA*.

(1) Most processes that can visibly change trust in a given store take place during *Sale finalization* (see Figure 1). It involves, among others, payment and delivery of products. Here, obviously, we can utilize knowledge originating from any form of commerce. Delay in delivery, bad quality of product, incorrect product, etc. are all facts that will negatively affect trust in a given (e-)store. Obviously, positive experiences will result in an improved trust.

(2) When the *CA* is not admitted to negotiations (obviously it is the *BA* that is not admitted; this is just a compact expression that we will use). While, obviously, nobody likes to be rejected, this is also an indication that according to a given shop our past behavior resulted in such a reduction of trust that we are no longer allowed to participate in negotiations. Since we use trust model with amnesia, over time we will be allowed back to negotiations. Thus, we do not consider this to be of crucial importance from the point of view of *CA*'s trust towards a store. Overall, this fact may be somewhat closer related to image than to trust.

(3) Separately, we should consider the fact that repeated price negotiations that result in relatively high prices will affect of image a given store. Obviously, it should not be assumed that high prices must result in a negative image of a store. There exist customers who purchase products from in-flight catalogs and for whom price is not an extremely important issue. At the same time, it is clear that (a) prices materialize in a different part of the system than *Sale finalization*, and (b)

information about them is easy to collect (and utilize) as most price negotiation mechanisms inform *all* negotiation participants about the winning price.

Let us now consider how the above described scenarios affect the behavior of the *CA*. Let us assume first, that for each store that a given *CA* interacted with in the past, it has computed a its trust-value. After *User-Client*'s request is formulated, the *CA* obtains from the *CIC* a list of stores that sell a given product. This list is then adjusted (see the statechart diagram of the *CA* presented in [3, Figure 2]). Specifically, the *CA* analyses the list of stores and updates their trust-values (recall that we use trust with amnesia and therefore trust has to be adjusted based on time that passed since last interaction). Then it checks if there are any stores with trust-value below a "threshold of trust." Such stores are considered untrustworthy and removed from that list. Obviously, when only very few stores remain on the active list, the *CA* may decide to adjust the threshold value vis-à-vis a given (unpopular) product and as a result to increase number of stores that it will try to interact with. One of the reasons for such a decision may be to obtain a broader perspective on valuation of thought after product within the marketplace.

However, in the opposite situation, when after removal of untrustworthy stores, the remaining number of shops carrying a given product is large, the *CA* may apply multicriterial ranking of stores (e.g. based on work of Saaty [19]) and select a smaller number of them to contact. Here the fact that a given store is being recognized as expensive may be used as one of possible criteria. However, since in the Saaty method each individual criterion is weighted by the user, for some users high price will be an important factor, while for some it will not. In other words, one can visualize this process as application of *User-Client* "profile" where all of her preferences (like: I prefer more expensive – high quality product rather than less expensive products; or I am looking for cheaper products rather than fast delivery) are used to weight various store-features and to obtain a final ranking.

One may ask: why not to send *BA*s to all possible stores? The answer is also related to trust relationships. As it is described below, each time when a *BA* wins a price negotiation, but does not complete a purchase, its trust-value decreases. Therefore, in the simplest scenario (when each *BA* participates only in a single price negotiation – and thus *BA*s do not re-enter negotiations), if *N BA*s are send out, and win price negotiations in *K* stores, then after making a purchase in one store trust will be reduced in *K-1* stores (and increased in that single store). Obviously, the larger the number of winners *K,* the larger the number of stores where the trust-value will decrease. Since decrease in trust-value may result in not being admitted to further negotiations, it becomes important to act judiciously and avoid trust reduction across a large number of stores.

Furthermore, when a very large number of *BA*s wins price negotiations then time to complete thorough analysis of all returned offers may take too long for some reservations – they will expire before we know that they represented the best opportunity to buy (Saaty-type analysis involves $O(n^3)$ operations; where $n$ is the number of offers to consider). Since each expired reservation makes seriously reduces trust-value (see the next section) we have to avoid such a situation and either use a less thorough method, or keep the number of outgoing *BA*s small so that the decision-making module can complete full multicriterial analysis in a non-prohibitive time.

Finally, let us discuss a particular situation, which resulted in our re-designing a part of the system due to the *CA* trust considerations. The issue concerns a specific situation when the shop starts running out of items of a given product for sale. In the original system design, checking if there is a product available for sale was the last thing that the *GA* was doing before releasing the *BA*s into price negotiations. Since it is conceivable that at this stage there was no product available for sale anymore, the *GA* had to make sure that this is not the case. Note that in our system lack of product available for sale may mean two things: (1) that there is no product at all, or (2) that all remaining units of that product are currently reserved (which makes it possible that there will be a product available for sale if one of existing reservations expires without a sale and thus the reserved item will be released back to the pool of items available for sale). In this case *BA*s were informed that there is no more product and we assumed that in this case they would self-destruct [3]. Obviously, such a situation could have affected *CA*'s trust in a store, as it could consider it inappropriate to stage *BA*s while knowing that price negotiations may never start due to the lack of products. As a result its trust in that store would decrease and could result in not sending *BA*s there in the future. To avoid such a situation we have decided to adjust the scenario. Let us assume that the current group of *BA*s (interested in product *X*) was send to price negotiations (and thus there is no agent waiting). In this situation, when the next *BA* interested in *X* arrives at the store, the *GA* will consult the *WA* if the project is actually available. If it is available, one unit is being pre-reserved (blocked), and the *BA* is accepted. In the case when there is no product at all, then the *BA* is informed that this is the case and rejected. In the case when no units are available, but some are reserved, the *BA* is informed about the situation and admitted to the pre-processing stage. However in this case it is not being provided with the *protocol* and *template.* It is only if a product is actually freed, that all *BA*s awaiting price negotiations (idle in the pre-processing stage) are being serviced further. In this way we have removed possibility that, due to the lack of product, the *CA* will reduce its trust in the *SA*. Since the *CA* was informed about the situation it cannot hold the *SA* responsible for cancelling negotiations altogether, at a later time.

Unfortunately further development (and implementation) of most of above described ideas, underlying trust between the *CA* and e-stores in the system, requires existence of fully working *Sale finalization* module. Thus far, in the development of the system our attention was focused on its remaining parts and this module remained somewhat neglected. Therefore we omit further discussion of *CA* trust and focus our attention on how *SA* trusts incoming buyers (and thus *CA*s they represent).

As indicated above, the first contact between the shop and the incoming buyer agents is facilitated by the *Gatekeeper agent.* It is the *GA* that informs about the decision to admit or not to admit a given *BA* to the store (to allow it to participate in price negotiations). This decision is based on the trust-value that the *GA*s (or, more specifically, the *SA*s) associates with a given *BA* (or, more specifically, *CA*). If the trust-value falls below a certain threshold (individual for each store) then the *GA* will not admit a given *BA*. Note that here only trust (not image) is taken into consideration. Let us now broaden our perspective somewhat and conceptualize events that have positive and negative impact on the image of a given *CA*. In what follows we will use in the narrative the *BA*, but in each case we should keep in mind that the *BA* represents a given *CA*. Therefore the trust relationship holds between the *SA* and the *CA*, while they are represented by the *GA* and the *BA*.

In the case of trust relationships viewed from the perspective of the *SA* it is the product reservation model used in our system that plays a crucial role. Recall, that if a given *BA* succeeds in price negotiations, it does not mean that it will buy a product. Instead, the *SA* (1) creates a reservation for a specific time, and (2) removes a given item from the pool of products available for sale. Therefore, for as long as the reservation is unexpired, this item cannot be sold to someone else. Since it is the role of the *SA* to sell as many products as possible, if the *BA* actually makes a purchase, its trust-value will increase (*CA* that makes purchase is a valued customer).

What happens if the *BA* does not make a purchase? There are two possible scenarios. (1) The *BA* relatively early during the reservation informs that it will not be making a purchase. Let us say that the total reservation time was 20 minutes, while the decision about abandoning purchase came within first 2 minutes. In this case, while being relatively "unhappy" with the *BA*, the *SA* can be somewhat understanding. It is important to keep in mind that in our system all sides "agreed" to the specific model of price negotiations and reservations. Therefore they have to accept consequences of such an approach. Here, it has to be obvious that there will be frequent sale cancellations and *BA*s that cancel order early during reservation should not be overly punished. (2) The reservation expires. In this case the *SA* has a "right" to be very "unhappy" as not only the product was kept out of the pool of products available for sale but we have to keep in mind that it is also possible that the next *BA* that did not win the negotiation could have actually purchased the product. Obviously, in this case our trust in a given *BA* will be substantially reduced. Note that as a result it may fall below the threshold and prevent the *BA* from being (re-)admitted to negotiations when it tries to access them the next time (see [3] for more details about re-admittance to negotiations).

Observe that what is particularly damaging is the case of repeating offenders. Let us assume that the *BA* immediately re-enters price negotiations and wins them for the second time, and again does not make a purchase due to an expired reservation. Such a *BA* may be considered to be a rouge agent (also dubbed a spoiler – see for instance results presented in [8] about spoilers in price negotiations) and thus our trust in that *BA* will be reduced considerably. Each subsequent offense of this type will result in an even larger penalty and may very fast result in an agent falling below the negotiation admittance threshold.

Let us also consider further scenarios that negatively affect *SA*'s trust in a given *BA*. First, when the *BA* enters the negotiation host and does not report as ready to participate in price negotiations. In our system [3], *BA*s are equipped by the *GA* with the *negotiation protocol* and the *negotiation template* (see also [6] for more details about price negotiations model used in our system). In the next step the *BA* should request its *strategy* from its *CA*. It is only after the *strategy* arrives and the *BA* is completely assembled, when it reports to the *GA* as ready to negotiate. It is easy to imagine that a *BA* may be staying at the host without reporting to negotiations. This could be caused by the delay in obtaining its *strategy* module. However, it could be also a result of a deliberate action (e.g. the *BA* could be doing something entirely different and steal local resources). Furthermore, it is also possible to envision that this could be a form of a distributed denial of service attack, where *BA*s flood a host and sit there without entering price negotiations. At this stage of our work, we do not want to deal with such sophisticated scenarios. Nevertheless, we see a *BA* not reporting that it is negotiation-ready for an extended period of time as a case of resource stealing and resulting in decrease of trust-value. Furthermore, we assume that such an agent will be killed by the host as soon as it stays past a certain time limit. However, let us note that this action should not be viewed as retaliation but rather as a simple act of cleaning the host from zombie-agents (e.g. *BA*s that cannot communicate with their *CA*s as they went down and cannot be reached anymore).

We can also envision situation, when something happens during *Sale finalization*. Let us say that the *BA* succeeded during price negotiations, confirmed purchase, but its *CA* did not pay (for one reason or another – due to the *CA* "negligence" – the payment process failed). In this case, obviously, trust in that *CA* will also be reduced. However, since as indicated above, our *Sale finalization* process has not been implemented in a robust way, we will omit this situation from further considerations.

### 4.3 REWARDS AND PENALTIES

Let us now assume that we have established trust-values for all agent pairs that are in trust relationship and specified procedures to increase and decrease it according to the scenarios described above. We can now summarize and consider in more detail what is the effect of trust and its increase/decrease on the interactions between the *BA/CA* and the *GA/SA* (please note that we omit here trust of the *CA* in the *SA* – its only effect is the *CA* deciding to not to send its representatives to the given *SA*). Let us assume that the system is already running for some time and *SA*s have already established trust-values for all *CA*s that they have interacted with so far. In this case:

(1) The most direct effect of the trust-value is the decision about admission to price negotiations. It the trust-value is not high enough – and the threshold value varies from a store to a store – the *BA* will not be admitted to price negotiations. Let us note that *BA*s may attempt at being admitted to price negotiations multiple times while attempting at purchasing a single unit of a product. Except the case when previous price negotiations resulted in a failure, each attempt at re-admission will be done with a different level of trust of the *SA* towards the *BA* (each successful price negotiation not followed by a purchase results in reduction of trust-value).

(2) Level of trust is directly related to the length of the reservation. The higher the level of trust, the longer the reservations time allowed by the *SA* (in this way *SA*s reward their best clients). Note that the longer the reservation time, the better is the situation of the *CA* as it has more time to establish which offer is actually the best. At the same time, the longer the reservation time, the longer is the time that the product is frozen and cannot be sold.

(3) Time to stay and "do nothing" before being removed from the system is also directly related to the trust-value. As the level of trust increases, the more "benefit of the doubt" is given to the *BA* (the longer it can stay without being killed).

## 5. CALCULATING TRUST-VALUE

Let us now present our initial proposal as to how the trust-value can be calculated in the above described system. Recall, that we have restricted our attention to the *SA* evaluating incoming *BA*s (and their master *CA*s). However, the proposed solution can be also utilized in the case of an opposite relationship. Let us introduce a *permanent trust-value function* $T_n(x) \in [\,-1,\ 1\,]$, where $n = 0, 1, 2, \ldots, +\infty$, denotes subsequent actions of a *CA* (denoted as $x$) that have been evaluated by the *SA*. We assume that initially for every *CA* ($x$) the trust-value $T_0(x) = 0$. Here, 0 should be understood as a "neutral" trust-value, -1 denotes complete distrust, while 1 means complete trust.

Let us now consider a trust-adjustment function *TA* specified for each interaction $e$ between the *CA* and the *SA*. Here, $TA(e) \in [\,-\alpha,\ \alpha\,]$, and $0 < \alpha < 1$. Events perceived as *positive* will have a *positive* $TA(e)$, *negative* events *negative* $TA(e)$, while *insignificant* events $TA(e)$ close to zero. Each interaction $e$ results in trust adjustment:

$$T_n(x) = (1 - |TA(e)|)T_{n-1}(x) + TA(e).$$

It follows from the above definitions that $T_n(x) \in (-1, 1)$. The sense of parameter $\alpha$ is as follows: the closer $\alpha$ is to 1, the shorter is the "memory" of *SA* about bad/good behaviors of *CA* and vice versa; the closer $\alpha$ is to 0, the longer the "memory" of *SA* about bad/good behaviors of *CA*. Therefore, parameter $\alpha$ can be viewed as a way in which *SA*'s memory is "managed" – the narrower the interval of values available for the *TA*(*e*) to take, the longer the overall *SA* memory. Function *TA*(*e*) is evaluated only after an interaction *e* takes place. For example, if winning *CA* confirms the making a purchase in time, the *SA* evaluates this behavior accordingly. Note that, this function allows us to facilitate trust-amnesia (which is one of the assumptions of our approach). When the event *e* is lack of interaction for a specific time, it is possible to set up the *TA*(*e*) in such a way to move the trust-value towards the neutral trust (closer to 0). We assume here that, since positive events (purchase) are rare vis-à-vis negative events (abandoning existing reservation), positive trust will be decaying more slowly than the negative image improving (though both converge to 0).

## 5. CONCLUDING REMARKS

In this paper we have discussed trust management in an agent-based e-commerce system. Our goal was to specify situations in which level of trust between interacting agents influences outcome of their encounter, as well as how various events taking place in the system influence the level of trust. As it turned out, currently in our system, while we do not have tools to fully explore the notion of reputation, we can design and implement a trust management system for *Shop agents* evaluating *Client agents* and adapting their behaviors on the basis of a history of interactions. Complete analysis of possible trust-affecting scenarios was followed by a proposal of a trust-value function that should match well with characteristics of our system.

One of crucial steps will be empirical evaluation of actual utility of the proposed trust-value function. We will attempt at performing appropriate tests as soon as the newly re-implemented system skeleton will be completed.

## REFERENCES

[1] ABDUL-RAHMAN A., HAILES S., *Supporting Trust in Virtual Communities*. In Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, 4-7 January 2000,

[2] AUDUN J., ROSLAN I., BOYD C., *A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems* (to appear)

[3] BĂDICĂ C., BĂDITĂ A., GANZHA M., PAPRZYCKI M. *Developing a Model Agent-based E-commerce System.* In: Jie Lu et. al. (eds.) E-Service Intelligence - Methodologies, Technologies and Applications, Springer, 2006 (to appear)

[4] BĂDICĂ C., BĂDITĂ A A., GANZHA M., IORDACHE A.,PAPRZYCKI M *Implementing rule-based mechanisms for agent-based price negotiations.* In: Proceedings of the 21st Annual ACM

Symposium on Applied Computing, Dijon, France, ACM Press, New York, NY, 96-100, 2006
[5] BĂDICĂ C., GANZHA M., PAPRZYCKI M., *Rule-Based Automated Price Negotiation: an Overview and an Experiment.* In: Proceedings of International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland. Lecture Notes in Artificial Intelligence, Springer-Verlag, (in print)
[6] BARTOLINI C., PREIST C., JENNINGS N. R., *A Software Framework for Automated Negotiation.* In: Proceedings of SELMAS'2004, LNCS 3390, Springer Verlag, 2005, 213-235
[7] BLAZE M., FEIGENBAUM J., LACY J., *Decentralized trust management.* In Proceedings of the IEEE Symposium on Research in Security and Privacy, number 96-17, MAY 96.
[8] BRANDT F.,WEI G., *Antisocial agents and Vickrey auctions.* In Proc. of the 8th Workshop on Agent Theories, Architectures and Languages, pages 335{347, August 2001.
[9] CAHILL V., SHAND B., GRAY E., et al. *Using Trust for Secure Collaboration in Uncertain Environments.* Pervasive Computing, 2(3):52–61, July-September 2003,
[10] CARBONE M., NIELSEN M., SASSONE V.. *A Formal Model for Trust in Dynamic Networks.* In Proc. of Int. Conference on Software Engineering and Formal Methods, Brisbane, September 2003
[11] GAWINIECKI M., GANZHA M., KOBZDEJ P., PAPRZYCKI M., BADICA C., SCAFES M., POPA G. *Managing Information and Time Flow in an Agent-based E-commerce System,* In: Proceedings ISPDC'2006 (to appear)
[12] *Jena 2 - A Semantic Web Framework,* Hewlett Packard, http://www.hpl.hp.com/semweb/jena2.htm
[13] JØSANG A., *Trust-Based Decision Making for Electronic Transactions.* In: L. Yngström and T. Svensson, editors, Proceedings of the 4th Nordic Workshop on Secure Computer Systems (NORDSEC'99). Stockholm University, Sweden, 1999,
[14] JØSANG A. *A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 9(3): 279–311, June 2001
[15] MANCHALA D.W., *Trust Metrics, Models and Protocols for Electronic Commerce Transactions.* In Proceedings of the 18th International Conference on Distributed Computing Systems, 1998
[16] MEERSMAN R. et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 809–815, 2005. Springer-Verlag Berlin Heidelberg 2005,
[17] PAGE L., BRIN S., MOTWANI R., WINOGRAD T., *The PageRank Citation Ranking: Bringing Order to the Web*. Technical report, Stanford Digital Library Technologies Project, 1998
[18] RESNICK P., ZECKHAUSER R., *Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System.* In M.R. Baye, editor, The Economics of the Internet and E-Commerce, volume 11 of Advances in Applied Microeconomics. Elsevier Science, 2002
[19] SAATY T.L. *The Analytik hierarchy process,* RWS Publications, 1990
[20] SCHAFER B.J. KONSTAN A, J, AND RIEDL J. *Recommender Systems in E-Commerce*. ACM Conference on Electronic Commerce (EC-99), November 3-5, 1999, Denver, CO.,
[21] WITHBY A., JØSANG A., INDULSKA J. *Filtering Out Unfair Ratings in Bayesian Reputation Systems.* In Proceedings of the 7th Int. Workshop on Trust in Agent Societies. ACM, 2004.
[22] YU B., SINGH M.P. *An Evidential Model of Distributed Reputation Management.* In Proceedings of the First Int. Joint Conference on Autonomous Agents & Multiagent Systems. ACM, July 2002.
[23] YU B.,SINGH P. M. *A social mechanism of reputation management in electronic communities*. In Proceedings of Fourth International Workshop on Cooperative Information Peers, 154–165, 2000.