Trust-based Contract Nets

Rino Falcone, Giovanni Pezzulo, Cristiano Castelfranchi, Gianguglielmo Calvi

Istituto di Scienze e Tecnologie della Cognizione – CNR - viale Marx, 15 – Roma
{r.falcone,g.pezzulo,c.castelfranchi}@istc.cnr.it

Abstract. In this paper we use a contract net protocol in order to compare various delegation strategies. We have implemented some different agents, having a set of tasks to delegate (or to perform by themselves); the tasks are performed by the agents in a dynamic environment, that can help or worse their activity. The agent rely upon different strategies in order to choose whom to delegate. We implemented three classes of trustiers: a *random trustier* (who randomly chooses the trustee whom delegate the task to); a *statistical trustier* (who builds the trustworthiness of other agents only on the basis of their previous performances); a *cognitive trustier* (who builds a sophisticated and cognitively motivated trust model of the trustee, taking into account its specific features, its ability and motivational disposition, and the impact of the environment on its performance). Our experiments show the advantage of using *cognitive representations*.

1 Introduction

The necessity of considering risky situations in open multi-agent systems (situations in which cooperation among different autonomous entities is not assured by the mechanisms and protocols of interaction) requires to model decision making tools into each single agent (to rely on other entities). In this context, trust building mechanisms are becoming an important subject of study. In fact, different theoretical and practical approaches [4, 5, 7, 10] have been analyzed and developed in the last years. This paper shows the first significant set of results coming from a comparison among different strategies for trusting other agents using a Contract Net protocol [2].

We introduced three classes of trustiers: a *random trustier*, a *statistical trustier*, a *cognitive trustier*. All the simulations were performed and analyzed using the cognitive architecture AKIRA [9].

The results show the relevance of using a *cognitive representation* for a correct trust attribution. In fact, a cognitive trustier perform better of a statistical one even when it has only an approximate knowledge of the other agents' properties.

In § 2 we describe the experimental setting; in § 3 we describe in details the experiments and we provide many data; in § 4 we discuss the experiments; in § 5 we conclude and propose some further improvements.

2 Experimental Setting

We implemented a Contract Net with a number of trustier agents that delegate and perform tasks in a variable environment. Each agent has to achieve a **set of task** and is defined by a set of features: **ability set**, **willingness**, **delegation strategy**.
- *Task set* contains the tasks an agent has to achieve; it has the possibility either to directly perform these tasks, or to delegate them to some other agents.
- *Ability set* contains the information about the agent's skills for the different tasks: each agent has a single ability value for each possible task; it is a real number that ranges in [0, 1]. At the beginning of the experiment these values are randomly assigned to each agent on each possible task.
- *Willingness* represents how much the agent will be involved in performing tasks (e.g. how much resources, will or amount of time it will use); this modulates the global performance of the agent in the sense that even a very skilled agent can fail if it uses not enough resources. Each agent has a single willingness value that is the same for all the tasks it tries to perform; it is a real number that ranges in (0, 1).
- *Delegation strategy* is the rule an agent uses for choosing which agent to delegate the task (e.g. random, cognitive, statistical). It is the variable we want to control in the experiments for evaluating which trustier performs better.

Agents reside in an **environment** that changes and makes the tasks harder or simpler to perform. Changes are specific for each agent and for each task: in a given moment, some agents can be in a favorable environment for a given task, some others in an unfavorable one. For example, two different agents -performing the same task- could be differently influenced by the same environment; or again, the same agent performing different tasks in the same environment could be differently influenced by this environment in performing the different tasks. Influences range in (-1, 1) for each agent for each task; they are fixed at random for each simulation. The environment changes randomly during the simulations: this simulates the fact that agents can move and the environment can change. However, for all experiments, if a task is delegated in an environment, it will be performed in the same one.

2.1 Delegation Strategies

In the Contract Net, on the basis of the offers of the other agents, each agent decides who to delegate [8] depending from its delegation strategy. We have implemented a number of different agents, having different *delegation strategies*:
- a ***random trustier***: who randomly chooses the trustee whom delegate the task to. This kind of trustier has no a priori knowledge about: the other agents, the environment in which they operate, their previous performances. There is no learning. This is used as baseline.
- a ***statistical trustier***: inspired to a number of works, including [3], assigns a mayor role to learning from direct interaction. It builds the trustworthiness of other agents only on the basis of their previous performances, without considering specific features of these agents and without considering the environment in which they performed. It is one of the most important cases of trust attribution; it uses the previous experience of each agent with the different trustees (failures and successes) for attributing to them a degree of trustworthiness that will be used for select the trustee in a new future

interaction. There is a training phase during which this kind of trustier learns the trustworthiness of each other agent as a mean value of its performances (number of failures and successes) on the different tasks in the different environments; during the experimental phase the statistical trustier delegates the most trustful agent (and continues learning, too). There is no trustier's ability of distinguishing how the properties of the trustee or the environment may influence the final performance.

- a ***cognitive trustier***: following a socio-cognitive model of trust [6,7] this kind of trustier takes into account both the specific features of the actual trustee and the impact of the environment on its performance. In this implementation there is no learning for this kind of agent but an a priori knowledge of the specific properties of the other agents and of the environment. It is clear that in a realistic model of this kind of agent, the a priori knowledge about both the internal properties of the trustees and the environmental impact on the global performance should not be perfect. We did not introduce for this kind of agent a learning mechanism (we are planning to do it in a future work) but we introduced different degrees of errors in the knowledge of the trustier that corrupt its perfect interpretation of the world. The cognitive model is built using Fuzzy Cognitive Maps (FCMs) [1] in which the specific features of the trustees (ability and willingness, both depending from the specific delegated task) and of the environment (in which the trustee will perform the task) are used as input parameters.
- ***best ability trustier***: who chooses the agent with the best ability score.
- ***best willingness trustier***: who chooses the agent with the best willingness score.

These two kind of cognitive agents can be viewed as having different "personalities".

2.2 The Contract Net Structure

We have performed some experiments in a *turn world*, others in a *real time world*.
In the turn world the sequence is always the same. The first agent (randomly chosen) posts its first task (*Who can perform the task* τ*?*) and it collects all the replies from the other agents (*I can perform the task* τ *in the environment w*). All data given from the offering agents are true (there is not deception) and in particular the cognitive trustiers know the values of ability and willingness for each other agent (as we will see later, with different approximations). Depending from its delegation strategy, the trustier delegates the task to one of the offering agents (in case, even to itself: self-delegation). The delegated agent tries to perform the task; if it is successful, the delegating agent gains one *Credit*; otherwise it gains none. The initiative passes to the second agent and so on, repeating the same schema for all the tasks for all the agents. At the end of each simulation, each agent has collected a number of *Credits* that corresponds to the number of tasks that the agents it has delegated have successfully performed. We have introduced no external costs or gains; we assume that each delegation costs the same and the gain of each performed task is the same. Since the agents have the same structure and the same tasks to perform, gained Credits are the measure of success of their delegation strategy.

In the *real time world* we have disabled the turn structure; the delegation script is the same, except for no explicit synchronization of operations. This means that another parameter was implicitly introduced: *time* to execute an operation. Collecting and analyzing messages has a time cost; agents that have more requests need more time in order to fulfill all them. In the same way, agents who do more attempts in performing a task, as well as agents who reason more, spend more time. In *real time*

world time optimization is another performance parameter (alternative or together with *Credits*), and some alternative trust strategies become interesting: in real time experiments we introduced another strategy:
- the ***first trustful trustier***: it is a variant of the cognitive trustier and it has the same FCM structure; but it delegates to the first agent whose trust exceeds a certain threshold: this is less accurate but saves the time of analyzing all the incoming messages. More, if some busy agent accepts only a limited number of tasks, or if there is a limited time span for performing the maximum number of tasks, it is important to be quick in delegating them.

2.3 Performing a Task

When an agent receives a delegation, it tries to perform the assigned task. Performing a task involves three elements: two are features of the agent (task specific ability and willingness), the third is the (possible) external influence of the environment. In order to be successful, an agent has to score a certain number of hits (e.g. 3); a hit is scored if a random real number in (0, 1) is rolled that is less that its ability score. The agent has a number of tries that is equal to ten times its willingness value, rounded up (i.e. from 1 to 10 essays). The environment can interfere with agent's activity giving a positive or negative modifier to each roll (so it interferes with ability but not with willingness). If the number of scored hits is sufficient, the task is performed; otherwise it is not.

In the simulations we used *3 hits as a default*; however, all the effects are stable and do not depend from the number of hits. We have performed experiments with different number of hits, obtaining similar results: choosing higher values leads to less tasks performed on average, but the effects remain the same.

This kind of task performing highlights both the role of ability and willingness (persistence).

2.4 FCMs for Trust

For the sake of simplicity we have assumed that each cognitive agent has access to all true data of the other agents and of the environment; these data involve their task specific ability, their willingness and their current environment. All these data are useful for many strategies: for example, the *best ability trustier* always delegate to the agent with the higher ability value for that task.

The *cognitive trustier*, following the socio-cognitive model of trust [6,7], builds an elaborated mind model of all the agents; this is done with Fuzzy Cognitive Maps, as described in [1][1]. The values of three nodes (ability and willingness as internal factors and environment as external factor) were set according to agent knowledge. The values of the edges reflect the impact of these factors and are always the same in the simulations. It has to be noticed that we never tried to optimize those factors: the results are always significant with different values. An additional point: while in the experiments the environment modifies the ability, in the "mental representation" of FCMs this is not the case: this is not an information that an agent is meant to know;

[1] With respect to the general model, for the sake of simplicity we assume that Unharmfulness and Danger nodes are always 0, since these concepts have no semantic in our simulations.

what it knows is that there is an external (positive or negative) influence and it aggregates it fulfilling the cognitive model. Figure 1 shows an (un-initialized) FCM.

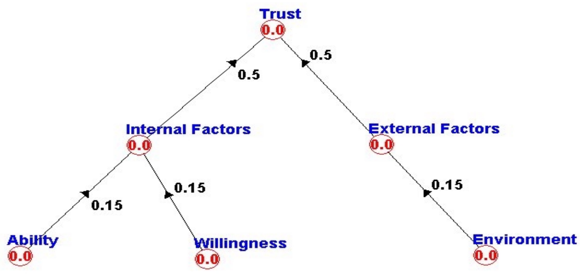

Figure 1. The FCM used by the *cognitive trustier*

3 Experiments Description

The first aim of our experiments is to compare the *cognitive trustier* and the *statistical trustier* in different situations: their delegation strategy represents two models of trust: derived from direct experience vs. built upon a number of cognitive features. The random strategy was added as a baseline for the difficulty of the setting. The *best ability* and *best willingness* strategies are added in order to verify in the different settings which are the single most influential factors; as it emerges from the experiments, depending from some parameters their importance may vary.

In all our experiments we used exactly six agents (even if their delegation strategies may vary); it is important to use always the same number of agents, otherwise the different sets of experiments would not be comparable. In each experiment the task set of all agents is always the same; their ability set and willingness, as well as the environment influence, are randomly initialized.

The experiments are performed in a variable environment that influences (positively or negatively) the performance of some agents in some tasks, as explained before.

In order to allow the *statistical trustier* to learn from the experience, all the simulation sets were divided in two phases (two halves). The first phase is meant for training only: the statistical trustier delegates several times the tasks to all the agents and collects data from successful or unsuccessful performance. It uses these data in order to choose whom to delegate in the second phase (in fact, it continues to learn even in the second phase). The delegation mechanism is always the same: it chooses the agent that has the best ratio between performed and delegated task; this number is updated after each result following a delegation. In order to measure the performance of this strategy, we analyzed only experimental data from the second phases.

The first experiment (**EXP1**) compares the *random trustier* (RANDOM), the best *ability trustier* (ABILITY), the *best willingness trustier* (WILLINGNESS), the *statistical trustier* (STAT), and other two cognitive strategies that differ only for how much they weight the environmental factor: *no impact* (TRUST) does not consider the environment, while *low impact* (TRUST_L) gives it a low impact (comparable to the other factors). In Fig.2 we show 250 simulations for 100 tasks.

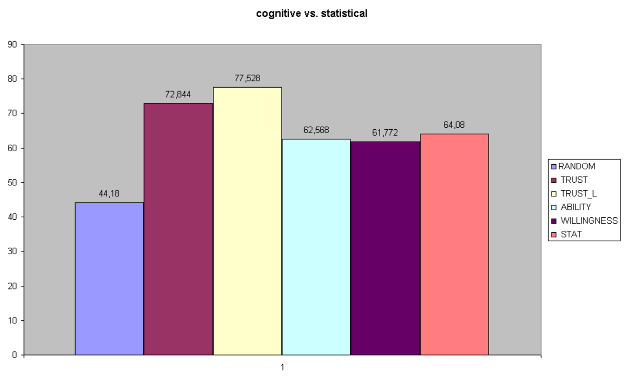

Figure 2. Experiment 1: comparison between many delegation strategies, 3 hits

We can see that the cognitive strategies always beat the statistical one[2]. Moreover, it is important to notice that recognizing and modeling the *external components* of trust (the environment) leads to very high performance: the cognitive trustier that does not consider the environment (TRUST) beats the statistical one (STAT), but performs worse than the cognitive trustier that gives a role to the environment (TRUST_L).

We have performed three more experiments in order to verify another interesting condition about learning (250 simulations, 100 tasks). Sometimes it is not possible to learn data in the same environment where they should be applied. For this reason, we have tested the *statistical trustier* letting it learn without environment and applying its data in a normal environment (**EXP2** - positive and negative influences as usual), in an always positive environment (**EXP3** - only positive influences), and in an always negative environment (**EXP4** - always negative influences). As easily foreseeable, the mean performance increases in an always positive environment and decreases in an always negative environment; while this is true for all strategies, the statistical strategy has more troubles in difficult environments.

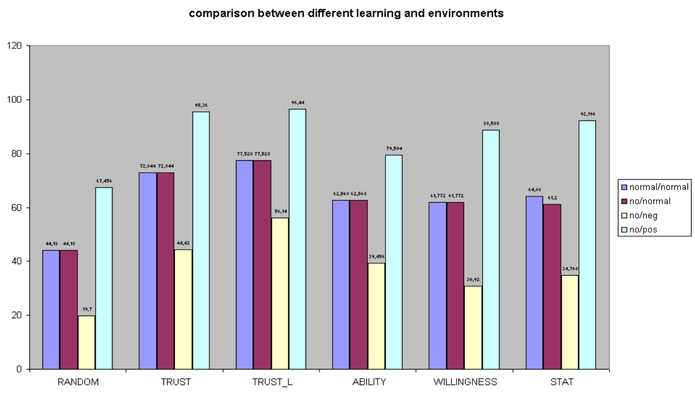

Figure 3. Experiments 1, 2, 3 and 4 compared

---

2 The results are similar e.g. with 5 hits (250 simulations, 100 tasks): RANDOM: 26,24; TRUST: 57,08; TRUST_L: 61,43; ABILITY: 40,58; WILLINGNESS: 48,0; STAT: 49,86.

Fig. 3 shows four cases:
1) learning in normal environment, task in normal environment (normal/normal);
2) learning without environment, task in normal environment (no/normal);
3) learning without environment, task in negative environment (no/neg);
4) learning without environment, task in positive environment (no/pos).

3.1 Using Partial Knowledge: The Strength of a Cognitive Analysis

The results achieved in the above showed experiments are quite interesting but even rather predictable. More interesting and with high degree of difficulty of prediction is the experiment in which we try to individuate the level of approximation in the knowledge of a cognitive trustier about both the properties of other agents and of the environment. In other words, we would like give an answer to the questions: when it is better perform as a cognitive trustier with respect to a statistical trustier? What level of approximation in the a-priori knowledge is necessary in order that this kind of trustier will have the best performance? For answering this interesting question we have made some other experiments (as **EXP5**) about errors in evaluation. As already stated, all the values we assume about cognitive features are true values: each agent knows all the real features of the others.

This is an ideal situation that is rarely implemented in the real world/system. In particular, in a multi-agent system environment there can be an evaluation process that is prone to errors. In order to evaluate how much error the cognitive trustier can deal with without suffering from big performance losses, we have compared many cognitive trustiers introducing some different levels of "noise" in their data.

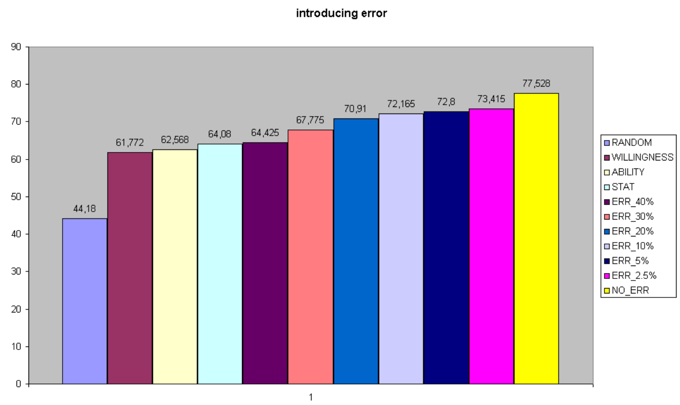

Figure 4. Experiment 5: introducing noise

Fig. 4 shows the data for the *random trustier* (RANDOM); the *best willingness trustier* (WILLINGNESS); the *best ability trustier* (ABILITY); the normal *cognitive trustier* (NOERR), as well as some other *cognitive trustiers* (ERR_40, ERR_20, …) with 40%, 30%, 20%, 10%, 5%, 2.5% error; the *statistical trustier* (STAT). While all the experiments used a set of six agents, we present aggregated data of different experiments. We have ordered the strategies depending from their performance; it is easy to see that even the worst cognitive trustier (40% error) beats the statistical trustier. Under this threshold we have worse performances.

3.2 Real Time Experiments

We have performed some real time experiments, too. **EXP6** (in Fig. 5) involves three cognitive strategies in a normal environment (250 simulations, 500 tasks).

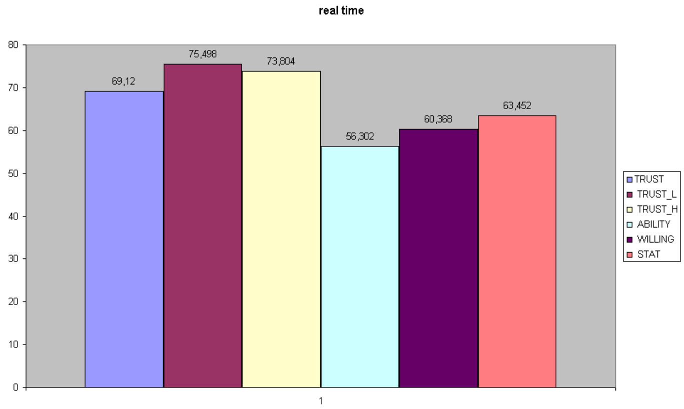

Figure 5. Experiment 6: real time

The differences between the cognitive trustier without environment and the two with environment are statistically significant; the difference between the two cognitive trustiers with environment are not. The results are very close to those that use turns; the differences depend from the limited amount of time we set for performing all the tasks: augmenting this parameter quicker strategies become more performing.

Another experiment (**EXP7**) aims at testing the performance of the first trustworthy trustier (FIRST). Here there are two parameters for performance: *Credits* and *Time*. *Time* represents how much time is spent in analyzing offers and delegating, i.e. how much offers an agent collects before choosing[3].

While in the preceding experiments the agents collected all the offers before deciding, here an agent can delegate when it wants, saving time. This situation is closer a real MAS situation, where agents act in real time and sometimes do not even know how many agents will offer help.

How much time is spent in delegation depends from the strategy and from simulation constraints. The random trustier can choose always the first offer it has, so it results to be the quickest in all cases. If there is a fixed number of agents and the guarantee that all them will offer, best ability, best willingness, the cognitive trustiers and the statistical trustier can build and use an ordered list of the agents: so they have to wait until the offer from the pre-selected agent arrives. In more interesting MAS scenario, without a fixed number of offering agents, each incoming offer has to be analyzed and compared with the others. In order to avoid waiting ad infinitum, a maximum number of offers (or a maximum time) has to be set.

However, in this scenario there can be other interesting strategies, such as the first trustful trustier, that does not wait until all the six offers are collected but delegates

[3] There are other possible parameters, such as time spent in reasoning or in performing a task. However, we have chosen only the parameter which is more related to the Delegation Strategy; the other ones are assumed to have fixed values.

when the first "good offer" (over a certain threshold) is met; this can lead to more or less time saved, depending from the threshold. Here we present the results of EXP7 (250 simulations, 100 tasks); in this case all agents wait for exactly six offers (and compare them) before delegating, except for the random trustier (that always delegates to the first one) and the first trustful trustier that delegates to the first one that is over a fixed threshold. Fig. 6 and Fig. 7 show the results for Credits (as usual) and Time spent (analyzed offers).

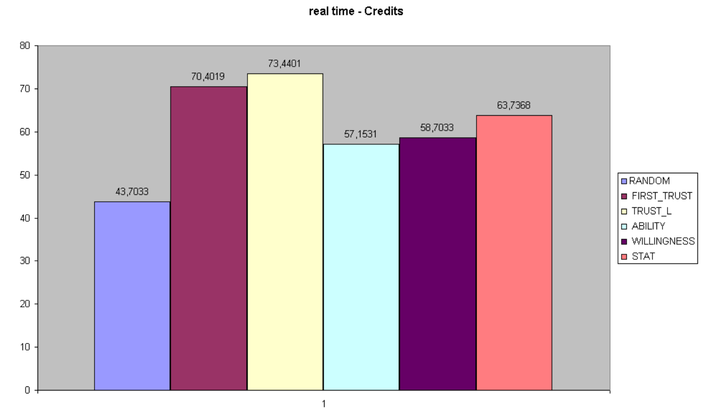

Figure 6. Experiment 7: real time, introducing the first trustworthy strategy

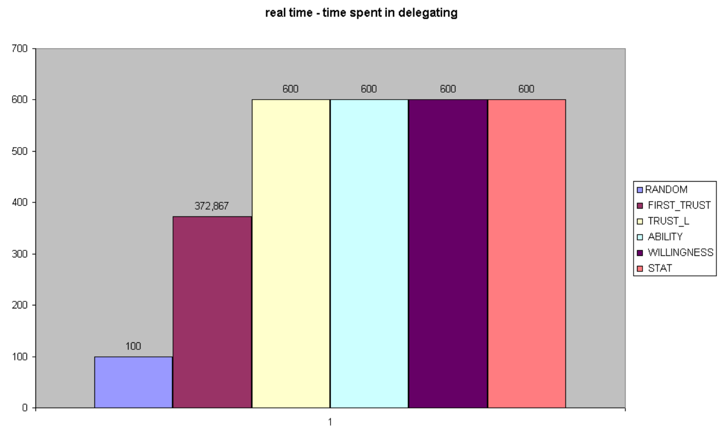

Figure 7. Experiment 7: time spent in delegating

The first trustworthy trustier still performs better that the statistical trustier, saving a lot of time. Depending from the situation, there can be many ways of aggregating data about Credits and Time. For example, in a limited time situation agents will privilege quickness over accurateness; at the contrary, in a situation with many agents and no particular constraints over time it would be better to take a larger amount of time before delegating.

4 Results Discussion

In our experiments we have tried to compare different trust strategies for delegating tasks in a contract net. The setting abstracts a simplified real-world interaction, where different Agents have different capabilities (mostly represented by Ability) and use more or less resources (mostly represented by Willingness) in order to realize a certain task. Very relevant is also the role of the environment because external conditions can make the tasks more or less easy to perform. On the basis of their trust, the delegating Agents (with different strategies) decide to whom to assign their tasks.
We analyzed two concepts of trust:
- the first (referred to the *statistical trustier*), that it is possible to model the trustworthiness of other agents only on the basis of the direct (positive or negative) experience with them; on the fact that there is only a dimension to consider: the number of successes or failures the agents preformed in the previous experiences.
- the second (referred to the *cognitive trustier*), based on a more complex set of factors that have to be considered before trusting an agent; in particular, both a set of trustier features and environmental features.

In all our experiments the cognitive trustiers perform better than the statistical one, both from the point of view of global successes (number of *Credits*) and stability of behavior (less standard deviation in the simulations data). The cognitive strategy better models the agents and environment characteristics and it allows to allocate the resources in a highly accurate way. Introducing a changeable environment does not decreases performance, providing that it is considered as a parameter; but even if it is not considered, the results are largely better that with statistical trustier.

The fact that an algorithm that knows the real processes implemented by the agents for achieving tasks uses a simulation mechanism of these processes for selecting the best performances is quite predictable. For this reason we have made new experiments introducing a significant amount of noise in the cognitive agent knowledge. The results show that the performance of the cognitive agent remains better than the statistical one till an error of 40%. So, the cognitive trustier is very accurate and stable under many experimental conditions. On the contrary, even with a large amount of data from learning (the training phase), the statistical strategy is not so performing. Moreover, if the learning is done in a different environment, or if the environment is particularly negative, the results are even worse.

With a low number of Hits (e.g. 3) the task is designed to privilege ability over willingness; however, augmenting the number of hits, the relative relevance changes. A strong environmental influence shifts the equilibrium, too: it modifies the ability scores, that become more variable and less reliable. Modifying the relative weight of those parameters (depending from the situation) into the FCM of the cognitive trustier can lead to an even better performance.

In the real time experiments, when implicitly time is introduced as an additional performance measure, a variant of the *cognitive trustier*, the *first trustful trustier*, becomes interesting: it maintains high task performance (measured by Credits) with a limited amount of time lost.

5 Comparison with Other Existing Models

Many existing Trust models are focused on reputation, including how trust propagates into recommendation networks [3,4,12,13]. On the contrary, our model evaluates trust in terms of beliefs about the trustee's features (ability, willingness, etc.); reputation is only one kind of source for building those beliefs (other kinds of source are direct experience and reasoning). In the present experimental setting there is not any reputational mechanism (that we could also simulate in the cognitive modeling), so a comparison with these models is not appropriate.

There are some other approaches where trust is analyzed in terms of different parts; they offer a more concrete possibility for comparison. For example, in [11], trust is splitted into: *Basic Trust*, *General Trust* in Agents, *Situational Trust* in Agents. Basic Trust is the general attitude of an agent to trust other agents; it could be related to our model considering it as a general attitude to delegate tasks to other agents in the trust relationships; in the showed experiments, we did not consider the possibility of introducing agents with the inclination to delegate to others or to do by themselves. In any case, the setting can certainly include these possibilities. General Trust is more related to a generic attitude towards a certain other Agent; the more obvious candidate in our setting is Willingness, even if the two concepts overlap only partially. Situational Trust is related to some specific circumstances (including costs and utilities, that are not investigated here); there is a partial overlap with the concept of Ability, that represents how well an agent behaves with respect to a certain task. So, the model presented in [11] is, to a certain extent, comparable with our one; however, it lacks any role for the Environment (more in general for the external conditions) and it introduces into trust the dimensions of costs and utility that in [6,7] are a successive step of the delegation process that is presented here in a simplified way.

6 Concluding Remarks and Future Work

Our experiments show that an accurate socio-cognitive model of trust allows to agents in a contract net to delegate their tasks in a successful way.

We plan to do a set of new experiments in which we allow even to the *cognitive trustiers* to learn from experience. While the learning of the *statistical trustier* is undifferentiated, the cognitive trustier is able to learn in different ways from different sources. In the cited socio-cognitive model for each trust feature there are four different sources: direct experience, categorization, reasoning, reputation; each of them gives a different contribute. More, higher level strategies can be acquired: for example, depending from the environment and the task difficulty (number of hits) an optimal weight configuration for the FCMs can be learned. In our simulations we assume that the costs (for delegation, for performing tasks, etc.) are always the same; for the future it would be interesting to introduce explicit "costs" for operations, in order to better model real world situations (e.g. higher costs for more skilled agents).

Another interesting field of exploration is the introduction of mechanisms for delegation monitoring; monitoring the performance of the delegated agents into the intermediate steps of the task, the delegator can decide for example to stop the delegation and to change delegee. In this case, a more complex cost model is needed, too (e.g. the costs of stopping or changing the delegation).

Moreover, we plan to include a reputation and recommendation mechanism, in order to add another trust dimension to the simulations. In this way we can introduce new delegation strategies (based on the existing systems in literature), and study how reputation interacts with the other cognitive features into the cognitive trustier.

7 References

[1] Falcone, R., Pezzulo, G., Castelfranchi, C. (2003). A fuzzy approach to a belief-based trust computation. *Lecture Notes on Artificial Intelligence*, 2631, 73-86.

[2] R.G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 1980.

[3] C. Jonker and J. Treur (1999), Formal analysis of models for the dynamics of trust based on experiences, *Autonomous Agents ‘99 Workshop on "Deception, Fraud and Trust in Agent Societies",* Seattle, USA, May 1, pp.81-94.

[4] S. Barber and J. Kim, (2000), Belief Revision Process based on trust: agents evaluating reputation of information sources, *Autonomous Agents 2000 Workshop on "Deception, Fraud and Trust in Agent Societies",* Barcelona, Spain, June 4, pp.15-26.

[5] A. Birk, (2000), Learning to trust, *Autonomous Agents 2000 Workshop on "Deception, Fraud and Trust in Agent Societies",* Barcelona, Spain, June 4, pp.27-38.

[6] R. Falcone and C. Castelfranchi, (2001), Social Trust: A Cognitive Approach, in C. Castelfranchi and Y. Tan (Eds), Trust and Deception in Virtual Societies, Kluwer Academic Publishers, pp.55-90.

[7] Castelfranchi C., Falcone R., (1998) Principles of trust for MAS: cognitive anatomy, social importance, and quantification, *Proceedings of the International Conference on Multi-Agent Systems (ICMAS'98)*, Paris, July, pp.72-79.

[8] Castelfranchi, C., Falcone, R., (1998) Towards a Theory of Delegation for Agent-based Systems, *Robotics and Autonomous Systems*, Special issue on Multi-Agent Rationality, Elsevier Editor, Vol 24, Nos 3-4, , pp.141-157.

[9] Pezzulo, G. Calvi G. (2004). AKIRA: a Framework for MABS. *Proc. of MAMABS 2004*.

[10] D. Gambetta, editor. Trust. Basil Blackwell, Oxford, 1990.

[11] S. Marsh. Formalizing Trust As a Computational Concept. PhD thesis, University of Stirling, U.K., 1994.

[12] A. Jøsang and R. Ismail. *The Beta Reputation System*. In the proceedings of the 15th Bled Conference on Electronic Commerce, Bled, Slovenia, 17-19 June 2002

[13] A Abdul-Rahman, S. Hailes. *A Distributed Trust Model*. New Security Paradigms. 1997