# Using Trust for Detecting Deceitful Agents in Artificial Societies *

Michael Schillo† Petra Funk‡
{schillo,funk}@dfki.de
Multi-Agent Systems Group
Saarland University/DFKI
Im Stadtwald
66123 Saarbrücken, Germany
Tel: ++49-681-302-2464
Fax: ++49-681-302-2235

Michael Rovatsos
rovatsos@knowbotic-systems.com
Knowbotic Systems Ltd.
Institute for New Media
Daimlerstraße 32
60314 Frankfurt a. M., Germany
Tel: ++49-69-941963-222
Fax: ++49-69-941963-178

**Abstract**

Trust is one of the most important concepts guiding decision making and contracting in human societies. In artificial societies, this concept has until recently been neglected. The inherent benevolence assumption implemented in many multi-agent systems can have hazardous consequences when dealing with deceit in open systems. Our aim is to establish a mechanism that helps agents to cope with environments inhabited by both selfish and co-operative entities. We achieve this by enabling agents to evaluate trust in others. We present a formalisation and an algorithm for trust so that agents can autonomously deal with deception and identify trustworthy parties in open systems.

Our approach is twofold: Agents can observe the behaviour of others and thus collect information for establishing an initial trust model. In order to adapt quickly to a new or rapidly changing environment, we enable agents to also make use of observations from other agents. We demonstrate the practical relevance of our ideas by means of a direct mapping from our scenario to electronic commerce.

*Abbreviated title:* Using Trust to Detect Deceitful Agents

## 1 Introduction

Multi-agent systems are typically of a complex and dynamic nature. Therefore, it is inherently impossible for a designer of such systems to pre-specify all behaviours

*To appear in: *Applied Artificial Intelligence Journal*, Special Issue on Trust, Deception and Fraud in Agent Societies, 2000.

†The author is funded by the Deutsche Forschungsgemeinschaft (DFG), contract number FI 420/1-1.

‡The author is funded by the European Commission: Platform project, contract number PL 97-2710.

and activities of a single agent or of a collection of concurrently active agents. Learning and adaptation offer elegant concepts and methods to solve these problems. Until recently, most multi-agent systems implicitly incorporated benevolence and co-operation as key features. With open systems and virtual co-habited worlds, such as electronic market places or other Internet forums, multi-agent systems evolve from pure collections of co-operating units to artificial societies enriched with social structures and attitudes. Socially intelligent agents, and their ability to cope with egoistic or betraying agents are new challenges arising within this development (Dautenhahn et al., 1997). Agents may find themselves confronted with deception and fraud; being able to reason about others' intentions before committing to actions becomes essential for survival in an artificial social setting.

Trust is one of the most important social concepts that helps human agents to cope with their social environment and is present in all human interaction (Gambetta, 1990). In artificial societies, such as those provided by state-of-the-art multi-agent systems, the real-world issues that arise from having selfish, antisocial, or unreliable members in the society are not adequately addressed. Although this issue has already been raised by Rosenschein and Genesereth (1986), most multi-agent systems are founded on benevolence assumptions, including assumptions regarding trustworthiness or reliability. With the growth of network services through the Internet, we find ourselves in hybrid artificial societies, where real-world assumptions and the whole range of possible behaviours must be taken into account.

A conceptualisation of trust and how trust can be used in artificial societies is a different subject of study than the techniques applied for secure network protocols and cryptography. They can guarantee identification of individuals and privacy of transmissions, but they cannot guarantee that an interaction partner has the competence she claims or that she is honest about her intentions. But the latter issues are essential for dealing with deception in such virtual worlds. Furthermore, determining trustworthiness by inquiring other agents is not an issue that is considered in research on data mining, information retrieval etc. as agents must expect their sources of information to possibly have motivation to bias their replies to such inquiries.

Much work has been done in how to identify strategies of agents and how to react to a variety of behaviours. In this area, the claim is that an agent needs to have a counter strategy for a range of possible behaviours at its disposal. Therefore, as proposed by Carmel and Markovitch (1998) a representation of the interaction partner's strategy is necessary (e.g. represented by a finite state automaton) in order to create an optimal reaction (the automaton that reacts optimally). Another concept is *reciprocity* as conceived by Sen (1996) and Biswas et al. (1999), where the response strategy of an agent is evaluated by thresholds that determine a level of accepted defective behaviour. The authors enable agents to evaluate a best response towards agents that they need to interact with, even if the strategies of these agents are unknown or known to be defective.

These approaches build on the success of Axelrod's (1984) experiments with round-robin tournaments of agent strategies in the Iterated Prisoner's Dilemma, where each strategy had to play against all other strategies (and to play against itself). Work in social psychology, however, has shown that in such games selecting the right strategy to decide *with whom* to play yields (at least in human groups) a better game performance compared to spending the same effort on *how* to play the game itself (Jin et al., 1993). Especially in the context of open systems like artificial and hybrid societies there is at least the chance of an alternative to every

interaction partner, so spending resources on consideration with whom to play is beneficial, if not crucial.

As a response to these issues, we propose in the following a conceptualisation of trust, which is established by an observation and communication process. Agents start out with no knowledge about the behaviour of other members in the society and then modify their model on trustworthiness of others according to observations and testimonies from agents that witnessed interaction behaviour. While interacting and observing, the model about other members of the society is refined and used to judge their reliability to commitments about future actions. Our approach includes two important information-gathering activities for the agent that is to identify trustworthy interaction partners: (i) collecting information about the trustworthiness or reliability of potential partners by observation and (ii) interviewing other agents about their observations. Testimonial evidence from interviews may be brittle, as witnesses may have different motives and may try to deceive agents about their true observations. Thus, every agent is confronted with noise in the information and also with the possibility that the source of information itself is biasing the data. Our formalisation of trust is able to deal with such not-so-trustworthy witnesses, enabling the agent to distinguish potential partners from undesired contracting parties.

Some of the ideas presented here have been inspired by the ideas of Bazzan and colleagues (Bazzan et al., 1997). They enriched Flood and Dresher's classical Prisoner's Dilemma game with agents with social attitudes. Players can be either rational egoists or generous altruists. This notion of moral sentiments was introduced to establish a method for altruistic agents to support each other. Their results demonstrate that pure rationality (attributed to the egoistic agents), in terms of never co-operating, does not pay off in the long run for neither the single egoist nor its social group. While Bazzan and colleagues provide their agents with knowledge about the structure of the society, we chose to adopt the paradigm *"The world does not come labelled"* (Edelmann 1987), i.e. our agents only know their own social attitude, they do not know that of other agents except for observed facts. In our scenarios the agents' activities are not strictly conform with their social attitudes; we introduce fuzziness into these social roles, by supplementing each agent with a probability factor for role conformity. We demonstrate how our agents use the formalisation of trust in order to deal with deceit by letting them interact in a game theoretic framework. It extends the Prisoner's Dilemma game by a pre-play partner selection phase and by a phase in which intentions can be announced before the game is actually carried out.

Within this framework we can study the importance of partner selection based on trust and can elicit deceitful announcements of intentions, while still maintaining compatibility to previous research in game theory. In our experiments agents negotiate about their potential partners for the game, based on testimonial evidence and their model of trust in others. During negotiation the agents are allowed to communicate with other agents to find out about their opinions on altruism and trustworthiness of other players.

This research is under development in the context of the recently created interdisciplinary field of socionics (Malsch et al., 1998). Socionics is concerned with studying the relevance of sociological concepts and theories for distributed artificial intelligence and vice versa. We have been strongly influenced by this endeavour and find the use of sociological concepts like trust, deceit, etc. considerably helpful to model artificial societies and automated contracting on behalf of human users.

The remainder of this article is structured as follows: In the next section we give more motivation for this work and for the use of the concept of trust. The environment for experiments with the disclosed Prisoner's Dilemma game and the TrustNet are presented in detail in Section 4. This is followed by a discussion of how this affects the behaviour of the agents and their adaptation. Then, in Section 6 we present an algorithm and an appropriate data structure to evaluate trust (TrustNet). We discuss experimental results in Section 7 and round up the presentation by concluding remarks and an outlook to future work in Section 8. In the following section we will explain why and to what extend we use social metaphors for our approach, and how they relate to learning and adaptation processes.

# 2 Social metaphors - Socionics

The already mentioned work of Bazzan et al. (1997) is an attempt to introduce terminology from the social sciences has already been mentioned. The authors tried to construct a society of more successful agents by providing the concepts of altruism, egoism and using altruists with moral sentiments to support themselves. Another important influence to this work is the newly created discipline of socionics, a recent effort initiated by the German sociologist Thomas Malsch (Malsch et al., 1998). In this effort, researchers intend to merge research from the two disciplines of Distributed Artificial Intelligence and sociology in order to obtain new insights to a number of problems that are common to both disciplines, e.g. the problems of distributedness, heterogeneity, emergence and distributed problem solving[1].

One of the first assumptions we rejected was the inherent benevolence assumption in multi-agent systems, a step already taken by Bazzan et al. In contrast to their work, we did not choose to replace this assumption by a set of fixed behaviours, but by fuzzy social attitudes. In our work, these attitudes can range from egoistic to altruistic, from honest to dishonest. As described before, our aim was to model artificial societies where the agents did not know about the attitudes of their opponents. So they must learn by observation in order to create reliable models. Also, we wanted to enable agents to adapt very quickly to a new environment. Thus, it is essential for them to learn from other agents that have encountered a similar situation before. However, intentions and motives can differ from agent to agent and an agent cannot hope that every other agent wants to co-operate. It may be the case that other agents will use their influence (*power*), when asked by others for their observations to try to manipulate them by communicating incorrect data (*betrayal* and *deception*).

Having outlined the problem in these terms, the solution is provided by social sciences in a rather straightforward manner: an agent has to learn which witnesses it can trust in order to provide it with meaningful data and to evaluate how far it can trust these witnesses. Note that this is not a problem about how accurate the data of the witnesses is, but of how willing they are to communicate it, be it accurate or not. Thus, in this case we are talking about the *trustworthiness* of the witness and not about the reliability of the data of the witness. Dealing with such brittle testimonies is a hard learning task that will end (if successful) in an adaptation of

[1]This scientific endeavour is reflected by a funding program of the Deutsche Forschungsgemeinschaft, the German equivalent to the NSF, that supports research on artificial societies and on the modelling of these

behaviour, so that the agent learns by observation and communication which other agents are co-operative partners and will therefore know which agents to seek and which to avoid in interactions.

The social science solution alone will not solve the problem in terms of algorithms and computer programs. Therefore we developed a computational model of trust and a new and superior way to calculate it than it has been done before (e.g. (Marsh 1994), (Maurer, 1996)). We achieve this with the aid of a graph-like data structure that stores the agent's observations and the testimonies that the witnesses communicated. Computing the model according to this data is then a process based on probabilistic assumptions which can be found in more detail in Section 5. Due to the complexity of the object of sociological research we had to use definitions of terms that may not always be the broadly accepted definitions in the field of sociology. However, using this terminology enables us to bridge the gap between sociology and Distributed Artificial Intelligence and helps us to describe our system in a more elegant way.

# 3 Disclosed Prisoner's Dilemma with Partner Selection

In order to evaluate our concept of trust, we chose an extension of the Prisoner's dilemma game enhanced with a partner selection phase (Schillo, 1999). Agents do not have any a priori knowledge about the social attitude of potential partners, but they can learn from experience and observation. We model egoistic and altruistic personality profiles, but we provide them with a fuzzy factor: each agent plays according to its social attitude with a given probability. This allows for the modelling of a range of agent behaviours, including the traditional "benevolent agent" by using the configuration 1.0 for both altruism and honesty and a malicious agent with value 0.0 for both variables. Isolation from future games is a (punishing) result from constantly playing defect. Since we focus here on the trust model used to finding partners and evaluating observations from the game, we use a standard pay-off matrix and do not go into the details of the game itself. Due to space limitations we would like to refer readers unfamiliar with Prisoner's Dilemma games to the literature (e.g. (Luce and Raiffa, 1957)).

The disclosed Prisoner's Dilemma with partner selection can briefly be described as a five-step process:

1. Each player pays a stake.
2. Pairs of players are determined by negotiation and declaration of intentions. Agents have the possibility to deceive others about their intentions.
3. The Prisoner's Dilemma game is played, bearing in mind the previously declared intentions.
4. The results are published. Due to limited perception, each agent receives only the results of a subset of all players.
5. The prizes are paid.

The first step is straightforward: agents dispose of a limited amount of points; if an agent loses all its points, it has to retire from the game.

For the second step, we introduce a contract net-like protocol (cf. (Smith, 1980)). The protocol is executed until each player has had a chance to find a partner. For

each new round, the agents are randomly sorted to guarantee equal chances to all players. The first agent in this list is chosen to be the manager (according to the protocol). It announces its intention for a game of Prisoner's Dilemma. All other agents answer whether they want to play with it and what their intention would be. The manager then may choose among them its partner for the game. If it has limited knowledge about the social attitudes of the bidders, it can make enquiries about them. Communication during this phase is strictly restricted in the following manner: the manager asks an agent $W$, the contents is the name of a bidder $Q$. $W$'s answer in turn is a list of observations that $W$ has made of $Q$. $W$ does not have to be honest in this communication. Therefore, the manager needs to evaluate the information gathered by these short interviews. If it has chosen a game partner, we allow this agent to communicate in the same fashion. This way the bidders are not disadvantaged by not having been a manager in this round.

In the third step the agents play the game. This is a conventional enactment of the Prisoner's Dilemma game. The agents do not have to play according to their announcements of phase two, a fact which constitutes a source of deception. The results of the games are published in step 4, enabling agents to observe others' behaviour in terms of frankness, reliability and trustworthiness. Each agent is only told the results of agents in its direct neighbourhood (the size of this neighbourhood, as well as the number of allowed interviews are variables of experiments are reported in (Schillo, 1999)). With this information the agent can then update and calibrate its TrustNet in order to have improved models for the next partner selection. In the final phase, step 5, the agents receive the prizes for their moves.

If an agent is not trustworthy in terms of sticking to its announced move, agents in a defined neighbourhood notice this. Therefore it can and will happen that agents may at first gain from abusing the ignorant members of an agent society. Later on we will demonstrate that we found in our experiments that after a number of rounds such agents are excluded from playing, because they are no longer trusted by the other members of the agent society. In the next section we show the connection between the described setting and real world applications, where trust and finding out whom to trust is a central issue.

# 4 Practical Relevance

We choose the *electronic commerce* application to demonstrate the practical relevance of our approach. In this particular field of interaction, as well as in our scenario, agents encounter other agents while interacting autonomously in order to maximise their performance. They may achieve this by co-operating with their contract partners in the long run, or try to make "fast cash" and exploit others. Additionally, agents can offer contracts (i.e. announce intentions to certain actions) that they may not exhibit. Electronic commerce is unbound by national frontiers and therefore free from national authorities.

National authorities as a means of stability are not available in many electronic commerce applications (and, hence, it will not be for some time for many business partners). In our setting this is reflected by the fact that agents will not be punished if they deceive their game partners. By supplying the concept of trust, we enable agents to track rapidly which agent is behaving in a deceptive way so that they still perform very well. To model the conflict between behaving co-operatively or deceitful and the respective outcomes, we use the Prisoner's Dilemma with the

addition that agents need to pay before they are allowed to join a round of the game. Agents will experience no gain in score if they do not find some one that will join them in a game. Thus, there is an indirect punishment to malicious behaviour, which in a human society might be described as *peer pressure*.

The idea of peer pressure is based on the natural assumption that there is some communication in the world about the players in the market, e.g. press, personal communication between business partners, etc. Again this communication is in both settings not objective. In our experiments, agents will to some extend be *betraying*. As they do not want to be found making up false data, they will try to bias the information they communicate. This can be achieved by leaving out the data they have observed that does not suit their intentions. We assume that they are motivated to make their competitors look *not* trustworthy and *not* suited for games with high pay-off in order to discourage players to choose other agents than themselves.

Electronic commerce and our setting have many features in common and they are a dangerous place for agents relying on benevolent contractors. Agents that succeed in our setting will therefore be interesting to investigate when looking at the real world application. Additionally, we currently envision two other applications of the proposed mechanism. We believe that in the future mechanism like "cookies" will be of more importance in the Internet world. The demonstrated solution will be an easy to implement way to determine which web-site you can trust to not abuse the information of the cookie set on your machine (e.g. for abusing knowledge about personal profiles). More serious are two similar applications where our approach is just as applicable: the problem of differentiating between friendly and malicious hosts or migrating programs in the research of mobile agents (e.g. (Sander and Tschudin 1998)) and the distributed management of public keys (e.g. (Maurer, 1996)).

# 5 Behaviour and Adaptation

The evolution of multi-agent systems from socially neutral to socially rich artificial societies offers new application domains, which are hosted by open systems such as the Internet. Electronic commerce is a broad application domain comprising severe traps to blindly trusting agents. The commonly suggested solutions of Trusted Third Parties as global authorizers requires mutual agreement by all designers of services and may therefore be hard to achieve. Other exogenous control methods, as have been described in (Armstrong and Durfee 1998), include the observation of the behaviour of others and communication of such observations. Our approach helps each single agent to establish a model of trustworthiness of other agents. With only few iterations, as will be shown in the results section, agents learn whom to trust and whom to exclude from future interactions. Our agents behave according to a probabilistic model. Depending on two parameters they are more or less likely to behave altruistic/egoistic or honest/dishonest. The modelling of other agents' behaviour lies in the task of approximating this parameter for each agent. This approximation is achieved by gathering data on past behaviour and then evaluating averages. The crucial point here is that our agents do not exclusively rely on their own observations, but they also take into account the testimonies of witness agents. As these may be betraying, every agent that receives data has to evaluate the behaviour of the witnesses. If there is not enough data on these witnesses, the

agent will recursively apply the mechanism described above to gather data on them, again using both, observations and testimonies. The behaviour adaption of the agents consists in the choice of their interaction partners. The proposed agents will not change their intended behaviour as specified by their honesty and altruism parameter. Instead they will adapt their behaviour in the sense that they decide not to interact with agents that behave in an undesired manner and rather try to interact with agents that co-operative more often.

# 6 The TrustNet

In the described setting agents will naturally aim at collecting maximum data on other agents in minimum time. An agent is aware of the fact that other agents may have a range of different behaviours of which only a few will be desirable to experience while interacting. As a consequence, counting on observations made by the agent himself is not sufficient. A solution to this problem is to competently use the observations of other agents (witnesses). However, these witnesses may try to deceive by communicating false information. In the following sections, we describe how an algorithm together with a data structure called "TrustNet" copes with the testimonies of witnesses when their trustworthiness can not be determined by observation alone.

## 6.1 Semantics of the TrustNet

During the game an agent has the chance to collect data of two types: the honesty with which a player announces what it will commit itself to and the way it chooses its options, altruistic or egoistic. Each agent stores this data in a graph where the nodes represent agents. Figure 2 is an example of a simple graph with only four nodes. New nodes for agents can only be added when adding an edge from an existing node to it (i.e. when another agent provides at least one item of information about this agent).

The nodes are annotated with two values, the trust, which the agent can put into the agent represented by the node and the altruism that can be expected from it. The edges carry information on the observations that the parent node agent told the owner of the net about the child node agent. This data comprises which round was observed and which behaviour of the target agent was observed in terms of honesty and altruism. The owner of the net is represented in every net as the root node (node $A$ in Figure 2). All other nodes are descendants from the root node, as this is the node that is the source of all information. Its outgoing edges resemble the observations it made. The other edges constitute information that has reached the owner via communication.

## 6.2 On the use of trust values

The essence of our approach to modelling trust is the insight that trust is not an event in the sense of probability theory but rather a degree of how high some peer's honesty is estimated. This seemingly trivial observation is overlooked by many approaches employing probabilistic methods to model trust. The purpose of probabilities is to express the frequency with which events occur and not to provide for a qualitative model of someone's behaviour (although the two are obviously

strongly connected). Starting from this observation, we briefly describe in this section why we oppose to the direct combination of trust values.

### 6.2.1 Common Mistakes in Combining Information from Witnesses

If an agent $A$ is to compute its trust $T(A,Q)$ towards an agent $Q$ numerically, it is desirable for $A$ to include any information that is communicated to $A$ by witnesses $W_1, W_2, \ldots, W_n$ about the honesty of agent $Q$. The honesty of $Q$ is the ratio $r$ between the number of rounds in which $Q$ enacted what it committed itself to and the number of total rounds. This means that $A$ receives a number of values $r_1, \ldots, r_n$ from witnesses and wishes to combine these to get an approximation $r'$ of $r$. Many approaches at this stage compute $r'$ as the average of $r_1, \ldots, r_n$, thereby neglecting the fact that the observations of $Q$'s moves are not independent, because the subsets of $Q$'s actions observed by $W_1, W_2, \ldots, W_n$ are not necessarily disjoint. This is a phenomenon that Pearl (1988) calls "correlated evidence" and to which he ascribes the effect that "[extensional systems] will produce the same conclusions whether the weights originate from identical or independent sources of information."

Figure 3 shows an example of such correlation that yields unreasonable results when evidence is combined: the round nodes represent the agents $A$ and $W_1, W_2, \ldots, W_n$, whereas the rectangles on the right hand side represent the individual actions of agent $Q$, which cause the witnesses to assign a certain honesty to $Q$ depending on the deceptive/truthful utterances of $Q$ that they have observed (those observations are shown by the unlabelled edges). The labelled edges to the left of the witnesses describe the trust values $T(W_i,Q)$ communicated to $A$. It can clearly be seen that taking $r'$ as the average of $T(W_1,Q)$, $T(W_2,Q)$ and $T(W_3,Q)$ yields $r' = 0.75$, a value twice as high as the actual $r = 0.375$, while $A$ would have been able to reconstruct a significantly more precise approximation if it only had used all the information that the witnesses had. Hence this way of combining the evidence seems highly unreasonable.

The solution to this problem is to uncover the actual observations that led to the individual $T(W_i,Q)$ values and to propagate values along causally dependent nodes as in Bayesian inference. Unfortunately, however, $A$ has no means to access the observations concerning $Q$'s actions towards $W_1, W_2, \ldots, W_n$ other than by what is said about them by the possibly deceitful witnesses. Therefore the full net structure cannot be constructed by $A$ in a reliable fashion and the correlations cannot be uncovered, so we show an alternative solution.

### 6.2.2 Trust and Transitivity

When using trust values obtained from a witness $W$ it is reasonable to believe $T(W,Q)$ only to the degree that we trust $W$, i.e. $T(A,Q)$ should depend on $T(A,W)$ as well as on $T(W,Q)$. At first glance it seems straightforward to make use of this transitivity, since most people would agree that "if Frank trusts Jack and Jack tells Frank "Jim is trustworthy" then Frank should trust Jim" is a sensible line of reasoning. However, "being trustworthy" is not a binary proposition (event). If it were, we could combine $T(A,W)$ and $T(W,Q)$ to compute $T(A,Q)$ by using transitivity and Bayes' rule:

$$P(T(A,Q)) = P(T(A,W) \cap T(W,Q)) = P(T(W,Q)|T(A,W))P(T(A,W)) \quad (1)$$

So if, e.g., $T(A,W) = 0.25$ and $T(W,Q) = 0.5$, then all that $A$ knows is that with a probability of 0.75 the actual trust value $W$ has towards $Q$ is different from 0.5 but

it is not obvious at all how this should affect $A$'s model of $Q$. To illustrate this fact, Figure 4 shows the case in which $A$ has made some observations about the behaviour of $Q$ itself (which yielded a direct trust value $T_d(A, Q) = 1.0$), and that it also has a model of how honest $W$ is due to $W$'s previous behaviour ($T_d(A, W) = 0.25$). In which way should $T(W, Q) = 0.5$ be taken into account when it is received by $A$, given that the observations on which $T(W, Q)$ and $T_d(A, Q)$ are based might overlap and given that $W$ may have lied about the real value of $T(W, Q)$?

The reader is encouraged to try any reasonably simple combination of weight-combining functions (such as max, min and arithmetic operations) to compute $T(A, Q)$ from $T(A, W)$, $T(W, Q)$ and $T_d(A, Q)$. We are quite confident to say that no such function will yield satisfactory results, because (as mentioned in 6.3.1) changing and combining local weights in such a causal system cannot capture all the correlations that lead to the behaviour of the variables. In Section 6.3 we describe our solution to these problem, based on the analysis of "betraying" as a stochastic process, a technique which we view as an adequate way of implicitly reconstructing witness observations in order to alleviate the problem of "correlated evidence".

## 6.3 Merging Information from Multiple Testimonies

There are two possible ways to collect information on other players. First of all, every agent has the chance to observe a number of players directly. The smart agent will use a second option: interviewing other agents. In the second phase of the described game protocol every agent chooses another agent to play with. It can select a partner from a number of agents, which offer to play with it. If the agent does not have enough information about them, it can request information from all agents it has met before.

Evaluating the information that an agent observed by itself is rather easy (we will call the agent who evaluates its trust in others the *decision-maker* henceforth). If he observed $n$ games of an other agent (the so-called *target agent*) and the target agent behaved honestly in $e$ games, the natural approximation of the behaviour would be that the target agent will be honest in the next game with probability $p = \frac{e}{n}$. But how can the decision-maker combine information of a witness with the degree of credibility of this source of information? And, more complicated still, how can the decision-maker combine the information gathered from $m$ agents and their corresponding $m$ trust values? As we argued in Section 6.2 we cannot simply average over their information. These agents that act as sources of information (which we have already named *witnesses*) may have only partial information on the target agent or bias the information they report.

The algorithm that we will now present deals with this by combining many sets of partial data, like a jigsaw puzzle, to approximate the big picture. Additionally, the algorithm deals with overlapping data sets. Also, the decision-maker has to deal with the fact that the witnesses may have the intention to lie about their observations and hide some information. The proposed algorithm deals with this by estimating how often the witnesses lied.

Before we present how this estimation is evaluated we will have to define *betraying* in this context. First of all, what could be the motivation for a witness to lie? Any agent will want to be the one who plays with an agent of highly altruistic behaviour. There are two ways of influencing other agents in the proposed game. First, every agent can try to make other agents appear less cooperative when being asked for

testimony on them. The decision-maker will then have a smaller tendency to play with them and chances are getting better that he may instead play with the witness. Second he can make other agents appear less trustworthy, so that the decision-maker will tend more and more to ask him and not others about testimonies, increasing his power. Therefore, we find it reasonable to assume that there is a motivation to betray in the sense that betraying about information on an agent consists of hiding a fact that would make the target agent look either altruistic or honest.

We denote information on the behaviour of an agent in a single game with $e$ if it was honest or $a$ if it was altruistic. In the following will only look at the calculations about honesty, since altruism can be dealt with analogously, as will be explained at the end of this section. The statement of a witness on the behaviour of a target agent is $\varepsilon$ if it does not have information or claims to have no information. The variable $p$ denotes the frequency of betraying. According to this, betraying is a function:

**Definition 1** The *Betraying* function.

$$Betraying : \{e\} \rightarrow \{e, \varepsilon\}$$

*where*

$$Betraying(e) = \begin{cases} e & \textit{with probability } p \\ \varepsilon & \textit{with probability } 1-p \end{cases}$$

This means that we expect witnesses to behave in the following way: when an agent requests information on a given target agent, it checks its own observations. It transmits all the data on dishonest and egoistic (non-altruistic) behaviour. It then looks up all data on honest behaviour. It applies the *Betraying* function to every item and transmits only those where the result of the function is $e$. Thus a witness will neglect information and not tell something that is not the truth. It will not say that a target agent has played dishonest in game $x$ if this was not the case. The reason for this is that the witness does not want to be uncovered by obviously betraying. The agent that requests the information may have observed game $x$ by itself. A less obvious alternative is to influence the judgement of others by providing biased information.

Mathematically speaking, what happens when this function is applied, is a so-called *Bernoulli-experiment*. It is like tossing a coin that will show heads with probability $p$. In this case, showing heads corresponds to "mentioning" the data item, tails is "hiding" it. Repeating this experiment is a *Bernoulli chain*. The traditional use of a Bernoulli chain is to determine the number of heads or tails that will show on tossing a coin $n$ times. We will do something different. After a witness has communicated some data, we know how many times the *Betraying* function has returned $e$, but we do not know how many times it returned $\varepsilon$. Or, in other words, we know the number of honest replies (which we also denote with $e$) but we do not know the total number of Bernoulli experiments $n$. An example of such a situation is given in Table 1. The first two rows of data represent the information from two witnesses. As the information from the witnesses comprises also the game number, the information can be collated to a result tuple, which eliminates the problem that the data from the two witnesses might be overlapping (see Section 6.2).

The decision-maker can assume that this data is correct, as it assumes that the agents do not want to be caught betraying. But how can it find out to what extend the witnesses have biased the reported data on game results? And if so, how much information has been hidden by them? We will now explain how exactly we

estimate the hidden amount of information, judging from the trust of the decision-maker into the witnesses. As soon as we have established this, data can be collated and evaluated for any number of witnesses on target agents or different length of paths in the TrustNet. It can be used by applying the evaluation recursively from the target agent through all its ancestors up to the root node (which represents the decision-maker itself and whose honesty can be regarded as being 1).
We use the following variables:

- $n$ is the amount of information the witness has on a target agent.
- $k$ is the estimate of the number of times the result of the *Betraying* function was $\varepsilon$, i.e. the amount of positive data on a target agent that the witness intended to hide.
- $e$ is the number of reported $e$ (where the result of the *Betraying* function was $e$).
- $p$ is the estimate the decision-maker has of the parameter in the witness' betraying function. This is either the result of the evaluation of its own observations or the result of a previous application of this algorithm.

By using the formula for binomial distributions it follows directly from this model that the probability for the event that the witness has lied $k$ times (if we replace $n$ by $k+e$) is:

$$P(T=k) = \frac{(k+e)!}{k!e!} \cdot (1-p)^k p^e \tag{2}$$

Now we can infer what the expectation value is, i.e. how often we can expect the witness to have lied. We need to distinguish two cases.

**Case** $e > 0$: If the witness reported some $e$, it follows directly that the expectation value $EX$ is:

$$EX = \sum_{k=0}^{n} k \binom{k+e}{k} \cdot (1-p)^k p^e = (k+e)p \tag{3}$$

Assuming that the witness has lied $k$ times ($EX = k$), this yields $k = \frac{ep}{1-p}$. In the case that it reports at least one positive information item, we consequently we have an approximation of the number of times the witness has betrayed by leaving out information. Now we take a look at the more difficult case in which the witness reports on zero positive information. This will occur if the witness lies very often (its $p$ is large) or if it has only very little positive information about the target agent. In any case we would like to infer something from its testimony. However, this is more difficult to achieve as the above evaluation would yield $k = 0$ for $e = 0$. We propose the following formula, to solve the problem of approximation more accurately.

**Case** $e = 0$: If we apply $e = 0$ to the binomial distribution equation, we get $P(T = k) = p^k$. In order to compute this value for a given $k$, we need to calculate the expectation value of this function. We will do this by determining the value $E$, where the area beneath the curve of the function to the left of $E$ equals half the whole area beneath the function. In mathematical terms, this means that the expectation value $E$ for $k$ is

$$\frac{1}{2}\int_0^\infty p^k \mathrm{d}k = \int_0^E p^k \mathrm{d}k \Rightarrow k = \frac{\ln\frac{1}{2}}{\ln p} \tag{4}$$

As we now know how to determine the amount of hidden information, we can fill in the data in the "$k$"-column of Table 1, i.e. we can now complete the table by

the number of $e$ that are missing for each witness. However, we still do not know how to merge this data for a number of witnesses. We could make assumptions, e.g. assume that the unreported $e$ have been observed in completely disjoint sets of games. Or assume that they have been observed in all the same games. Or we could take the average of the unreported $e$. As we have argued before, it is not acceptable to neglect the possible dependence of the data of the witnesses (see Section 6.2). We believe that the following heuristic is more reasonable.

We assume that if we look at all the data in the reported tuples, we find that it is randomly distributed over the games that were reported just as well as the data for the unreported part of the tuple. We conclude that the distribution of the unreported $e$ will be similar to the distribution of the reported $e$. In mathematical terms this means that the density of the data (the relation of the overlapping of the data in the tuples) may be assumed to be constant for the reported and the unreported data. The density of the reported data only depends on known variables. The number of witnesses is known, as well as the total number of all reported entries in all tuples. Also, it is clear that we can determine the length of the result tuple. This will provide us with the following equation for density:

$$density = \frac{matrixEntries/witnesses}{lengthOfTuple} \tag{5}$$

Assuming that the density is the same in the unreported data, the density will determine the length of the tuple that has to be added to the already existing result tuple. We know (judging from the motives of the witnesses) that we can fill up this additional space with positive data on the target agent. We already know the number of agents and using the above equations we can determine the total number of the tuple entries that we expect to be hidden by the witnesses.

$$density = \frac{hiddenEntries/witnesses}{additionalTupleLength} \tag{6}$$

Using the density equation for the reported data, this provides us with an equation to determine the additional length of the result tuple:

$$additionalTupleLength = \frac{hiddenEntries/agents}{density} \tag{7}$$

We now have an approximation of what the witnesses tried to hide from the decision-maker, i.e. how honest they were. Using this information and the accounts of the witnesses yields a better approximation of the behaviour of the target agent than was available before. However, the algorithm relies on the provision of honesty evaluations for the witnesses. These can be obtained by recursively traversing the TrustNet as it has by definition a root node (the decision-maker itself) whose honesty can be assumed to be 100%. The procedure for evaluating altruism is in close analogy. First, the honesty of the witnesses is evaluated as above. Then, the algorithm establishes an approximation of the altruism information the witnesses gathered from the information they reported. To achieve this, the information on altruism (number of $a$ reported) is used in place of honesty information $e$.

The quality of the approximation will increase with the quality of the estimate of the witness' honesty. So the overall performance of this calculation will increase with the amount of data that is available to the decision-maker, which is a desired behaviour of the algorithm. One of the advantages of this rather complicated procedure is that we avoid propagating weights on the trustability of witnesses through

the net, so we do not face the aforementioned problem of how to combine these weights. By reconstructing for every merge of information an approximation of what the witnesses would have said *if* they had been completely honest about their information, we gain intuitive and more precise models for witnesses and target agents.

# 7 Selected Results

Using social metaphors to describe learning in multi-agent systems does not only provide elegant models, it also delivers significant results in terms of performance increase. In this section we present a number of results[2] that have been produced by simulation experiments using the disclosed Prisoner's Dilemma with partner selection (see above) and a society of 40 agents in 20 different behaviour configurations that compete for highest pay-off. As mentioned before, our agents are characterised by two parameters: the probability for honesty/dishonesty and for altruism/egoism. We chose four different values for honesty (0.01, 0.33, 0.66 and 0.99) and five values for altruism (0.01, 0.25, 0.50, 0.75 and 0.99). We included every possible combination of these values, which results in the 20 different agent strategies (i.e. behaviour configurations). So this society is highly heterogeneous.

As platform for our simulations we chose the Social Interaction Framework (SIF) that has been developed at DFKI (Schillo et al., 1999). As a basis for our analysis we used the data of seven simulations series and over 29,000 hours of computation time, distributed over six Linux PCs and a SPARC Ultra 2. From the convergence of the data we can conclude that this set of data is representative for the setting and its configurations. In the simulations, we had two groups of agents, an experimental group that used the model of trust to evaluate possible game partners and a control group that only used its own observations for this evaluation. All configurations of behaviour appeared in both the experimental and the control group.

First, we take a look at some example data to get a flavour of the improvement of the performance of these agents. For this first analysis we chose agents that had a high degree of egoism and deception (Figure 5) and agents with high altruism and high honesty (Figure 6). These two configurations can be looked at as the two extremes in a real world scenario: a malicious host or contract partner (with altruism 0.25 and honesty 0.33) and a friendly mobile agent or electronic commerce customer (with altruism 0.99 and honesty 0.99). The cumulative pay-off in the disclosed prisoner's dilemma was used as performance measure. Figure 5 shows that in the short term the performance of the agents using the TrustNet (experimental group E 0.25 0.33) is superior to the corresponding control group (C 0.25 0.33) where agents only regarded their own observations.

The increase of performance of the experimental group is decreasing with the growing quality of the models of the control group agents and their growing intention not to play with agents with non-cooperative behaviour. The other figure shows analogous results for a group of altruists. The experimental group (E 0.99 0.99) is by far more successful than the control group (C 0.99 0.99) and after 25 games the surplus in performance is still growing. This surplus is due to the fact that the altruists from the experimental group do not let themselves be deceived as many times as the control group agents, which is shown by the graphs. It should

[2]The complete discussion of the results of this research is part of the thesis of Michael Schillo (1999).

be noted that these graphs are averages on several hundred conducted simulations.

To some extend, our groups seem to behave sub-optimally: the altruists in Figure 6 do not have a better performance than the egoists in Figure 5, an effect which is not desirable. Further simulations with societies of agents which all used the TrustNet (no control group) have shown that the reason for this is the social intelligence of the society as a whole. If there are only socially intelligent agents in the society, the performance of the egoists decreases (as they find less agents to deceive) and the performance of the altruists increases (as they find more agents which trust them to be altruists). This modified setting and the corresponding performances are shown by the graphs annotated with "no control group" in the two figures. These graphs show that the altruists can perform the better, in comparison to the egoists, the more socially competent agents exist in the society. They also show that if agents use the TrustNet they perform better than the control group, be they egoists or altruists. As agents cannot rely on the social intelligence of other members of the society, we continue analysing mixed societies.

When we look at the average quality of the models, we can observe that there is a significant increase. Figure 7 shows the average error of the behaviour models for the agents of the control and experimental groups respectively. With the decreasing error, the quality of the model increases. This increase is the cause for the aforementioned gain in performance. As the agents have learned faster and more about the behaviour of others, they choose their interaction partners more competently. In order to measure this, we calculated for each group of agents the error in the learned approximation compared to the actual behaviour of any other member of the society. Figure 7 shows that agents using the TrustNet have a model with 20% less variance already after round ten. This results in the need for the control group to learn eight rounds longer to reach a similar quality of their model.

This advantage is of major importance to the performance of the agents in the society. Figure 8 shows the overall increase of all agents of the experimental group compared to the control group in percent (for a detailed score gain discussion see above). During the first 50 interactions in a society of absolutely unknown configuration, the performance of the agents develops a significant improvement compared to the agents that just use their own observations. The total score gain of the agents in the experimental group is more than 15%, reaches a level of plus 5% after ten rounds and remains significant beyond the 50 interactions we examined.

# 8 Conclusion

## 8.1 Summary

With our concept of trust and its formalisation we have provided a mechanism that allows for a better adaptation of agents to the behaviour of previously not assessed peers. This is achieved by enabling agents to use data communicated by others, even if these witnesses can be assumed to be deceptive about their knowledge and are manipulating this data. With this feature, the amount of observation data is enlarged, which allows for a quicker and better approximation of features to an extend that justifies its application in settings like electronic commerce and the mobile agents problem.

We analysed material that shows how much the performance of agents can im-

prove when our approach is used. The result is that agents can learn models from scratch almost twice as fast as other agents who use their own observations only, while still reaching the same or better accuracy. They achieve this even when confronted with witnesses that communicate manipulated data. Furthermore, we enable co-operative agents to form groups and play among themselves and thus, when having worked out who belongs to this group, forming more and more stable groups of cooperative agents that profit from mutual support. In the course of developing this method, we have used social metaphors as a language for describing interaction in multi-agent systems. We consider this a natural approach, as we assume a close analogy between multi-agent systems (i.e. artificial societies) and human societies.

## 8.2 Future Work

Trust is a mechanism for creating trustworthy channels for communication and can by nature be used for communicating a range of features important to multi-agent systems. We intend to explore further applications including the use of other features than altruism or honesty and to evaluate the changes in agent performance. We will also investigate how quickly agents can adapt by using the TrustNet if they enter a society of agents, who already have learned about each other's attitudes. Furthermore, the effects of changes of the behaviour over time and especially in more complex forms of betraying that are guided by strategies deserve further analysis, especially the issue of how agents can adapt to such changes. Varying the ratio of egoists and altruists leads to different distribution curves on the agent society structure. Researching the effects of the TrustNet in such different society structures is another interesting challenge.

More generally, these steps taken into research towards modelling concepts from the social sciences and using them in the context of artificial societies seems very promising. This is not only the case because, as we have shown, they can be used to improve the performance of individuals in social contexts, but, much more importantly, because they enable us to develop notions of "social intelligence" that are indispensable when it comes to building distributedly intelligent systems.

# References

Armstrong, A., and E. H. Durfee. 1998. Mixing and Memory: Emergent Cooperation in an Information Marketplace. In Proc. Third International Conference on Multi-Agent Systems, ed. Y. Demazeau, Paris, France.

Axelrod, R. The Evolution of Cooperation. 1984. New York: Basic Books.

Bazzan, A. L. C., R. H. Bordini, and J. A. Campbell. 1997. Agents with Moral Sentiments in an Iterated Prisoner's Dilemma Exercise. In (Dautenhahn et al., 1997).

Biswas, A., S. Sen, and S. Debnath. 1999. Limiting Deception in Social Agent-Group. In Deception, Fraud and Trust in Agent Societies. Proceedings of the workshop at the Autonomous Agents Conference, ed. C. Castelfranchi, Y. Tan, R. Falcone, and B. Firozabadi, National Research Council, Institute of Psychology, Rome, Italy.

Carmel, D., and S. Markovitch. 1998. How to Explore Your Opponent's Strategy (almost) Optimally. In Proc. Third International Conference on Multi-Agent Systems, ed. Y. Demazeau, Paris, France.

Dautenhahn, K., J. Masthoff, anb C. Numaoka. 1997. Socially Intelligent Agents. Papers from the AAAI Fall Symposium, Cambridge, Massachusetts, Technical Re-

port FS-97-02.
Edelmann, G. 1987. Neural Darwinism: The Theory of neural group selection. New York: Basic Books.
Jin, N., N. Hayashi, and H. Shinotsuka. 1993. An experimental study of prisoner's dilemma networks: formation of committed relations among PD partners. Japanese Journal of Experimental Social Psychology. 3.
Luce, R. D., and H. Raiffa. 1957. Games and Decisions, Introduction and Critical Survey. New York: Wiley.
Mller, H.J., Malsch, T. and Schulz-Schaefer, I. (1998). SOCIONICS: Introduction and Potential. Journal of Artificial Societies and Social Simulation. vol. 1(3).
Malsch, T., M. Florian, M. Jonas and I. Schulz-Schäfer. 1996. Expeditionen ins Grenzgebiet zwischen Soziologie und Küstlicher Intelligenz. Künstliche Intelligenz 2: 6-12.
Marsh, S. P. 1994. Formalising Trust as a Computational Concept. PhD Thesis, Department of Computing Science and Mathematics, University of Stirling, Scotland.
Maurer, U. 1996. Modelling a Public-Key Infrastructure. In Proc. 4th European Symposium on Research in Computer Security, ed. E. Bertino, H. Kurht, G. Martella, and E. Monolito. Lecture Notes in Computer Science 1146, Heidelberg, New York, Berlin: Springer.
Pearl, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, California: Morgan Kaufmann.
Rosenschein, J. S., and M. R. Genesereth. 1985. Deals among rational agents. In Proc. International Joint Conference on Artificial Intelligence, 91-99.
Sander, T., and C. F. Tschudin. 1998. Protecting Mobile Agents against Malicious Hosts. In (Vigna 1998).
Sen, S. 1996. Reciprocity: A foundational principle for promoting cooperative behavior among self-interested agents. In Proc. Second International Conference on Multiagent Systems, AAAI Press, Menlo Park, CA, 1996.
Schillo, M. 1999. Trust and Deceit in Multi-Agent Systems. Diploma Thesis, Department of Computer Science, Saarland University (in German).
Schillo, M., J. Lind, P. Funk, C. Gerber, and C. G. Jung. 1999. SIF - The Social Interaction Framework. System Description and User's Guide to a Multi-Agent System Testbed. DFKI Research Report RR-99-02, Saarbrücken, Germany.
Smith, R. G. 1980. The Contract-Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Trans. Computers, 1104-1113.
Vigna, G. Mobile Agents and Security. 1998. Berlin: Springer-Verlag.

# Tables

| | *Data on game results* | | | | | | | | | | | | | $k$ | $n$ | $e$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Witness 1 | | | | √ | | √ | | | × | | × | × | | × | ? | 6 | 4 |
| Witness 2 | | √ | | √ | | | × | | × | × | | | | | ? | 5 | 3 |
| **Result** | | √ | | √ | | √ | × | | × | × | × | × | | × | ? | | |

Table 1: Example for the reported data of two witnesses.

# Figure Captions

Figure 1: The disclosed Prisoner's Dilemma with partner selection.
Figure 2: A simple TrustNet.
Figure 3: Combination of trust values from multiple witnesses.
Figure 4: Transitive trust estimation.
Figure 5: Egoist performance.
Figure 6: Altruist performance.
Figure 7: Variance in agents' models' errors.
Figure 8: Performance gain when using trust.

# Figures

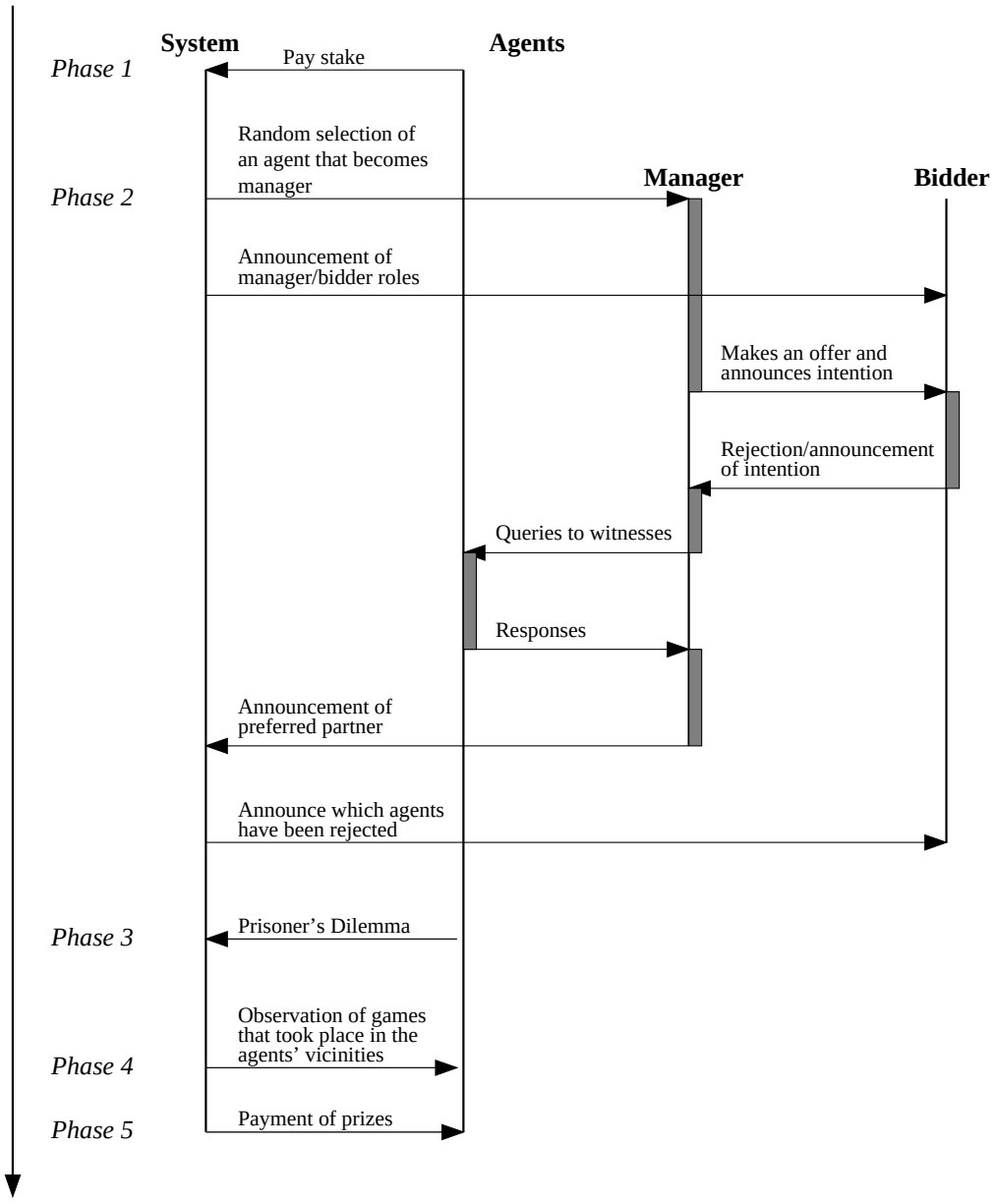
System
Agents
Phase 1
Pay stake
Random selection of an agent that becomes manager
Manager
Bidder
Phase 2
Announcement of manager/bidder roles
Makes an offer and announces intention
Rejection/announcement of intention
Queries to witnesses
Responses
Announcement of preferred partner
Announce which agents have been rejected
Phase 3
Prisoner's Dilemma
Observation of games that took place in the agents' vicinities
Phase 4
Phase 5
Payment of prizes

Figure 1:

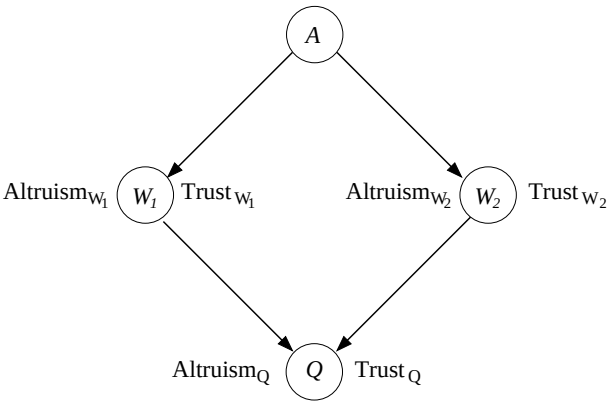


Figure 2:

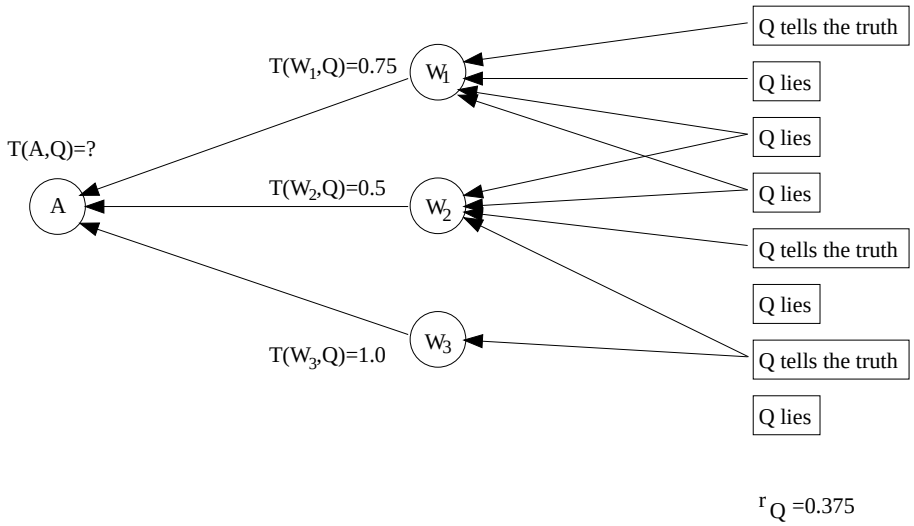


Figure 3:

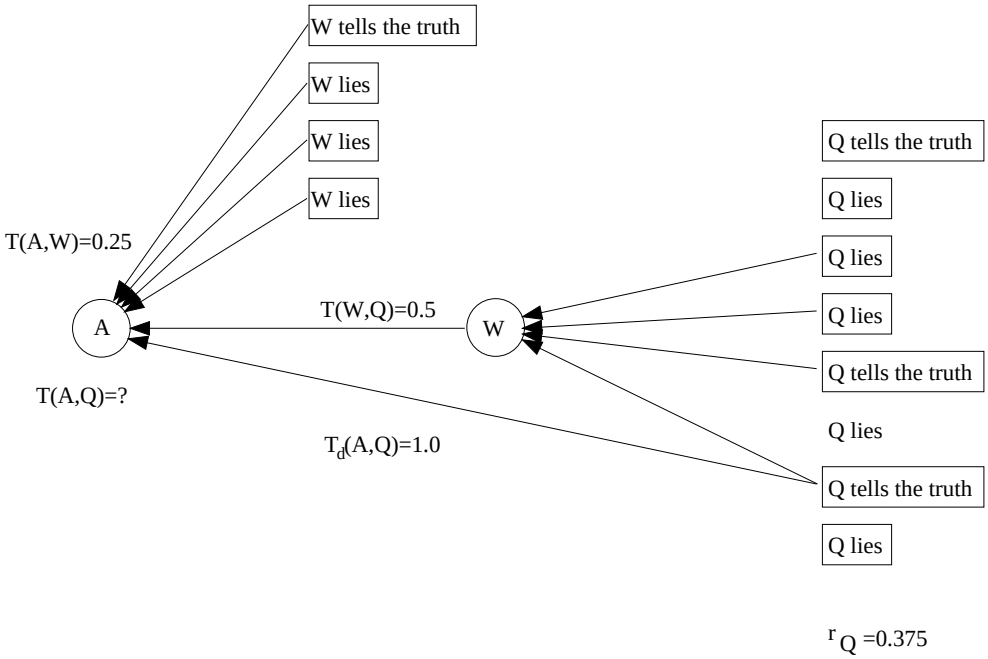


Figure 4:

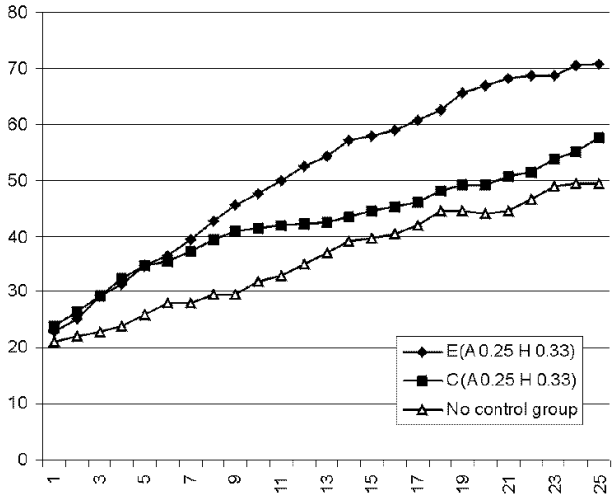


Figure 5: