Experiments in Building Experiential Trust in a Society of Objective-trust Based Agents

Mark Witkowski, Alexander Artikis and Jeremy Pitt

Intelligent and Interactive Systems Group
Department of Electrical and Electronic Engineering
Imperial College of Science, Technology and Medicine
Exhibition Road, London SW7 2BT
United Kingdom
`{m.witkowski, a.artikis, j.pitt}@ic.ac.uk`

Abstract. In this paper we develop a notion of "objective trust" for Software Agents, that is trust of, or between, Agents based on actual experiences between those Agents. Experiential objective trust allows Agents to make decisions about how to select other Agents when a choice has to be made. We define a mechanism for such an "objective Trust-Based Agent" (oTB-Agent), and present experimental results in a simulated trading environment based on an Intelligent Networks (IN) scenario. The trust one Agent places in another is dynamic, updated on the basis of each experience. We use this to investigate three questions related to trust in Multi-Agent Systems (MAS), first how trust affects the formation of trading partnerships, second, whether trust developed over a period can equate to "loyalty" and third whether a less than scrupulous Agent can exploit the individual nature of trust to its advantage.

1 Introduction

Software Agents are increasingly being required to make decisions and act locally, but also operate in the context of a "global" Multi-Agent Society (MAS). As these Agents become fully autonomous they become forced to make decisions about when and when not to engage (for instance to request information, to delegate important tasks or to trade) with other Agents. They must rely on internalised beliefs and knowledge about those other Agents in the society. This reliance on beliefs forms the basis of a *trust relationship* between intentional entities.

The trust relationship, in its broadest sense, has proved difficult to define [7], [8], [10], [15], [16], [17], [26]. We synthesise the following as a working definition, suited to the purposes of this paper. "*Trust is the assessment by which one individual, A, expects that another individual, B, will perform (or not perform) a given action on which its (A's) welfare depends, but over which it has restricted control*". Trust therefore implies a degree of dependency of A on B. This dependency may be reciprocal. Where the dependency relationship is asymmetric and one individual gains control over the other the relevance of the trust relationship is weakened for both A and B [16]. Equally, as the element of imposed compulsion in the relationship between individuals increases, the role of trust recedes. Similarly, the role of trust is reduced as the protagonists A and B acquire more complete information about each other (when they may accurately assess the future outcome of each transaction) [17]. Williams [26] summarises the trust relationship: "*agents co-operate when they engage in a joint venture for the outcome of which the actions of each are necessary, and where the necessary action by at least one of them is not under the immediate control of the other*". The trust relationship may further be subject to exogenous events under the control of neither party, which may or may not affect the relationship [16].

Autonomous software Agents face all these issues, dependency on others, restricted control, incomplete information and the effects of exogenous events. It is little wonder, then, that the issues of trust between Agents should attract attention. Until an adequate system of compunction is widely adopted (through legislation, or by mutual agreement, for instance) this situation is likely to remain. Griffiths and Luck [11] emphasise the notion of trust as a reciprocal of risk, in the context of co-operative planning between Agents. Marsh [17] considers the risk/benefit relationship for Agents in a Distributed AI context. Castelfranchi and Falcone [7] divide the notion of trust estimation into component belief types that one Agent might hold with regard to another. They argue that such beliefs may be combined to form a *Degree of Trust* ($\mathbf{DoT}_{XY\tau}$) measure, which may in turn be used to decide whether a task of type (τ) should, or should not, be preferentially delegated by Agent X to some other Agent, Y. Jonker and Treur [15] present a formalised framework for the description of trust based on sequences of experiences between Agents.

The concept of trust within a society is closely allied to that of reputation ([3], [6], [28]). Reputation systems provide a mechanism by which individual Agents within their society can obtain information about other Agents without, or prior to, direct interaction and can lead to gains for the individuals and society as a whole [6]. We argue that trust should be based, whenever possible, on direct experience rather than on accumulated social attitude or shared reputation. As in real life, there is a limit to what can be achieved by wondering about what another entity might, or might not, do in any particular circumstance. We recognise that the definition of trust both as a function of accumulated beliefs and as a function of direct experience will be important to the construction of Agents and Agent Societies in the future. Such direct experiences can form an *objective trust measure* - the trustworthiness of another Agent put to the test and recorded as the basis of selecting that individual for future dealings. Such direct observation methods are important as they serve to ground in experience other assessed trust and reputation mechanisms.

In this paper we consider *objective Trust-Based Agents* (oTB-Agents), Agents that select who they will trade with primarily on the basis of a trust measure built on past experiences of trading with those individuals. The purpose of this work is to be able to investigate some important questions that arise when Agents are given a "free choice" as to whom they will co-operate with. This paper will consider three questions:

1) What happens when Agents who rank experiential trust and trustworthiness highly form into trading societies?
2) Does a trust relationship established between Agents over a period of time equate to loyalty between those Agents when trading conditions become difficult?

3) Trust, however it is evaluated, is personal; is it in an Agent's interest to appear trustworthy in some cases, and not care in others?

We take a practical and experimental approach in our investigations. To this end we present a scenario where Agents must choose who to trade with on the basis of a trust relationship developed between them and then adopt a concrete example within which to discuss and evaluate oTB-Agents. Section two provides an overview of that test domain. By describing the mechanism within this concrete example, we do not intend to convey any presumption that its application should be restricted to this or any particular application area, as we do not consider this to be the case. Section three defines the main functional components of an oTB-Agent. Section four briefly describes our Agent simulator and then presents the results of some experiments that shed light on the questions just posed. Finally, we discuss related and future work and draw some conclusions.

2 The Trading Scenario

In order to test this notion of objective trust we establish a simulated trading environment in which many individual Agents must select partners with which they will trade on an ongoing basis. This continued trading within a closed community allows trust relationships to be made, sustained or broken over an extended period. The trust relationship must be essentially symmetrical; each party must be able to behave in a trustworthy or untrustworthy way towards others, and have others behave similarly towards them. To complete these experiments the individual Agents must also be subject to various exogenous events (those beyond their control), which force them to act in an untrustworthy way towards certain trading partners. We adopt a specific example, which is described next.

Fig. 1 shows an idealised model for a telecommunications *Intelligent Network* (IN). The IN provides an infrastructure in which different types of Agent may form a trading community, as well as acting as an interface layer between end-user consumers of a communications service and the underlying telecommunications network which will transport voice and data information between geographically distinct points.

We consider two distinct Agent types in this paper. *Service Control Point (SCP) Agents* are associated with *Service Control Points*, access portals to the telecommunications network. *Service Switching Point (SSP) Agent*s serve *Service Switching Points*, providing access points for consumers of telecommunications services. There may be a large number of SCP and SSP Agents forming a single IN. Each SCP acts as an agent or broker for the suppliers of telecommunications bandwidth and is tasked with ensuring that the available bandwidth is sold. Conversely, each SSP acts as agent or broker for end-consumers of telecommunications services, tasked with ensuring that sufficient bandwidth is reserved to meet the needs of those consumers.

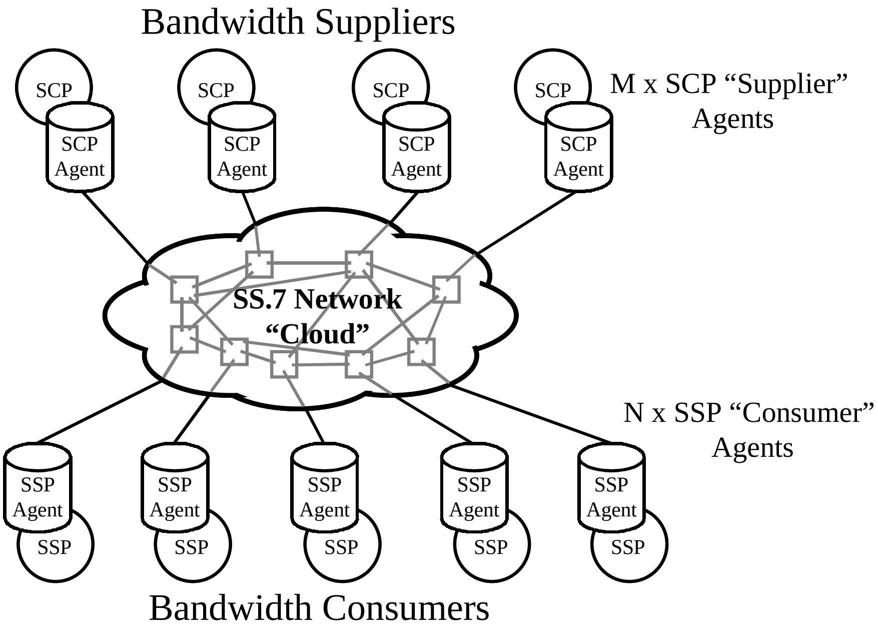

Fig. 1. Intelligent Networks Scenario

In this model message passing between SCP and SSP Agents is assumed to take place over an SS.7 network and to use a contemporary Agent Communication Language and Protocol, such as FIPA-ACL [9], [13]). Beyond requiring that the transmission of messages between Agents is timely and reliable, we will not consider related issues of inter-Agent communication further in this paper.

Agent Architecture models for IN management have proved to be rich ones for investigating issues directly related to resource allocation and load control in the context of current telecommunications systems ([13], [19], [20], [23]). Rather than concerning ourselves with issues relating to overall performance of the network, we will concentrate on the effects of trading decisions based on "objective Trust Based" (oTB) principles. We will focus on the performance of individual Agents from the perspective of the degree to which they trust, and are trusted by, other Agents in the society. In maintaining this focus on issues relating to trust we have developed a "trading scenario", which gives both SCP and SSP Agents the opportunity to behave in a trustworthy or untrustworthy way in their dealings with fellow Agents. This then forms a basis on which individual Agents select the Agents they will trade with in future.

2.1 The Trading Cycle

Trading is divided into equal time slots, called a *trading cycle*. At the beginning of each trading cycle every SSP (customer) Agent receives a demand for resource (bandwidth in the scenario) and makes bids to SCP (supplier) Agents to cover that demand. SSP Agents must select SCPs they trust to offer them the resource they require. If the SCP does not offer to cover an SSP's bid for bandwidth resource, then the SSP has reason to regard that SCP as untrustworthy. While demand may vary between trading cycles, the total amount of resource available is taken as fixed. Each SCP Agent must attempt to distribute its supply of resource to SSP Agents that it trusts to pass that resource on to its end-users. Any resource not taken up by SSP Agents is deemed lost, to the detriment of the SCP Agent. SSP Agents that fail to use resource offered to them are considered untrustworthy.

All SSP and SCP Agents each maintain a *trust vector*, recording the opinion the Agent holds about the trustworthiness of each of the other Agents with which it can trade. The trust vector forms the primary source for selecting trading partners, and is itself updated after each transaction.

Each trading cycle involves three transaction steps (each corresponding to an ACL performative between individual Agents). First, the *bid step*, in which SSP Agents receive their demand load and issue bids to SCP Agents to meet that load. Second, the *offer step*, in which SCP Agents make offers of resource in response to bids they receive. Third, the *utilisation step*, in which SSP Agents distribute the resource units they have been offered to their customers, and notify the SCP that offered the resource whether or not they utilised all the allocation they were offered.

2.1 The Bid Step

At each trading cycle every SSP Agent receives a quantity of demand from its customer base, which is the sum of their (the customers') estimates of the resource they require for the next trading cycle. Each SSP must then select one or more SCP Agents it trusts using an *allocator function*, and issue a *bid message* performative to them indicating the number of units of resource it requires. An SSP Agent may dishonestly (or perhaps prudently) *overbid* its requirement, thereby ensuring it will receive at least as much resource as it requires. In doing so it risks having to return unused units, and be seen as untrustworthy by the SCP Agent that reserved resource for it.

2.2 The Offer Step

Each SCP Agent receives a quantity of units bid from SSPs willing to trade with it. SCP Agents select which SSP bids it wishes to honour using a *quantifer function*, the choice being derived from the Agent's trust vector. The SCP Agents then communicate the offers of resource they are prepared to make back to the SSPs that made the original bids, the *offer message* performative. An SCP may not offer, in total, more resource units than it has access to. To do so would, in this scenario, introduce another round of transactions.

2.3 The Utilisation Step

Once an SSP Agent has received all the offer messages from SCP Agents, it will attempt to satisfy the customer demand for the current trading cycle from the offers of resource allocation that it has obtained. If it has received more resource than it requires it returns the excess to one or more SCP Agents on the basis of a *utilisation function*. Returns are notified to SCP Agents in a *utilisation message* performative. Also at this step the SSP Agent updates it trust vector using its *SSP trust function*, on the basis of the difference between the quantity the SSP Agent bid for against the quantity it received from SCP Agents. We assume accountability, in that an SCP Agent can meter units actually consumed at the request of an SSP Agent, so that an SSP cannot just request an unlimited number of units and just discard the excess (thereby appearing trustworthy to the SCP). On the other hand, each SSP is free to return unused units to any SCP, thereby managing its trust relationships.

Finally, on receipt of the utilisation messages, each SCP Agent can update its own trust vector according to its *SCP trust function* by comparing the quantity of resource requested against that actually utilised.

We treat the resource (bandwidth) as a true commodity. Any SSP may request resource from any SCP. We further treat the resource to be a fixed price item. Agents may not “spend more” to secure extra supplies in times of shortage, or reduce their prices in times of oversupply. When supply and demand are mismatched individual Agents must decide which Agents they will favour over others, this is at the heart of the “does trust beget loyalty?” question posed earlier.

There is no overall control or centralised mediation in this system model (as, for instance, in the auction model of Patel, *et al*., [18]). Each Agent makes its trading decisions based on its past experiences of trading with other Agents in its community, updating its trust vector, and so affecting its future decisions, based on each new transaction. In the model, Agents that do not adhere to the communications and transaction protocols are excluded from the trading arrangement. Messages sent inappropriately, such as an SCP offer where no bid was made, can be discarded and the sender considered “untrustworthy” for attempting to supply an unsolicited service.

3 The Allocator, Quantifier, Utilisation and Trust Functions

This section describes the SSP-Allocator, SCP-Quantifier, SSP-Utilisation and the Trust functions used by SCP and SSP Agents in detail. Together these five functions encapsulate the key components of oTB-Agents. Fig. 2 illustrates the internal structure of the SSP and SCP trading Agents used in the experiments to be described later, and indicates the order in which each of the five functions is invoked in the context of the overall trading cycle described in the previous section.

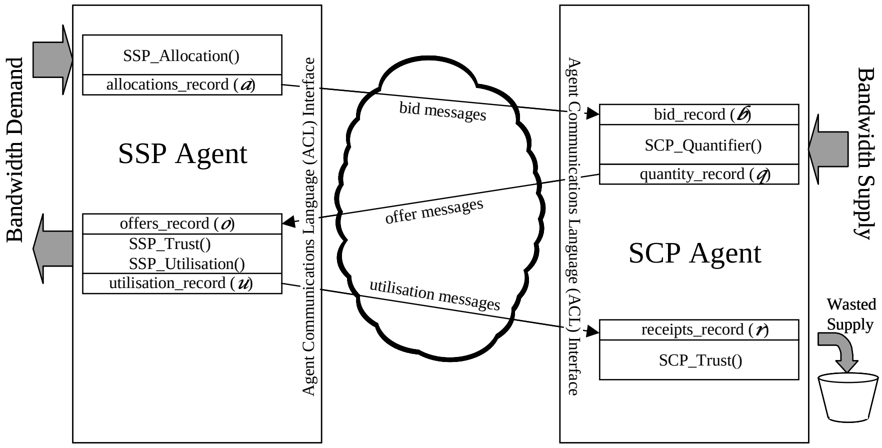

Fig. 2. The Trading Scenario

In a society of N SCP Agents trading with M SSP Agents, the trust vector owned by the nth SCP Agent will be represented by ${}_{n}t$; its trust rating of the mth SSP, a scalar value, by ${}_{n}t_{m}$. Conversely, the mth SSP's (${}_{m}t$) trust rating of the nth SCP Agent by ${}_{m}t_{n}$. Individual trust ratings are scaled from 0 (complete distrust) to 1.0 (complete trust). The use and management of these trust values is central to the operation of an oTB-Agent, they are the principal way in which other Agents are selected to trade with. The manner in which it is used, and the mechanism by which it is updated, define important aspects of an Agents apparent "personality" (the way it appears to other Agents) within the society. The *allocations record* (a), *offer record* (o) and *utilisation record* (u) are message buffers used by SSP Agents to prepare messages for sending (a and u), or receiving (o) messages from SCP Agents. The *bid record* (b, receive), *quantity record* (q, send) and *receipts record* (r, receive) are used by SCP Agents to buffer messages to and from SSP Agents. They employ the same indexing notation as t.

3.1 SSP Allocator Function

The SSP allocator function divides the total demand (`actual_demand`) received by an SSP Agent for the current trading cycle into smaller units and populates an *allocations record*, ${}_{m}a_{n}$, which holds the number of units of resource the SSP Agent m will be requesting from SCP Agent n.

The allocator function is controlled by three parameters. (1) The *overbid rate*, `obrate`, which determines how much extra resource the Agent will bid for above its actual demand. Overbid is expressed as a percentage. (2) The *split rate*, `srate`, which determines how many SCP Agents will receive bids from this SSP Agent. This effectively ameliorates the risk for the SSP Agent that any particular SCP Agent will refuse it supply. The split rate is expressed as an integer ≥ 1, but ≤ number of SCP Agents. (3) The *exploration rate*, `erate`, which determines the probability with which the Agent will ignore its trust ratings and send a bid to a random SCP Agent, where 0 represents no exploration of the market and 1.0 causes the SSP Agent to always select suppliers at random. The exploration rate parameter addresses a practical problem familiar in the reinforcement learning paradigm, that of balancing the advantages to be gained from trading with known and already trusted partners with the opportunity to discover better partners from the larger pool ([21]).

The allocator function is best described procedurally:

```
For each SSP Agent m do:
    1)  Clear ${}_{m}a$
    2)  Set demand ← actual_demand * obrate
    3)  Set bid_packet_size ← demand / srate
    4)  If (rand < erate) Set ${}_{m}a_{r}$ ← bid_packet_size
                where ${}_{r}$ is a randomly selected SCP Agent and
                rand is a randomly generated number, 0 .. 1.0
    5)  Else for SCP Agent x, where x is max(${}_{m}t_{x}$) and ${}_{m}a_{x} = 0$
        Set ${}_{m}a_{x}$ ← bid_packet_size
    6)  Repeat from step 4 until all bid packets allocated
```

Step 5 successively selects the most trusted, then the next most trusted until all the bid packets have been allocated. Once the allocator procedure is completed the SSP Agent issues a bid message to every SCP Agent where ${}_{m}a_{n} > 0$ (i.e. a bid has been allocated). Apart from the random selections, bids have been sent to the most trusted trading partners.

3.2 SCP Quantifier Function

The SCP quantifier function distributes the SCP Agent’s limited supply amongst all those SSP Agents that made bids, it does so on the basis of trust, as recorded in its trust vector. The function is unparameterised. Received bids are recorded in the *bid record* ${}_{n}b$, the SCP quantifier function populates the *quantity record* ${}_{n}q$, which records the offers to be made. If the total of bids (`total_bid_value`) received by the Agent total less than the available supply, the value of each bid is simply transferred to the quantity record, as all SSP Agent bids can be satisfied. When bids exceed supply the following procedure is invoked to distribute the available supply on the basis of trust:

```
For each SCP Agent n do:
    While total_bid_value > 0
    For SSP Agent x, where x is max(${}_{n}t_{x}$) and ${}_{n}q_{x} = 0$
    Set ${}_{n}q_{x}$ ← ${}_{n}b_{x}$ if total_bid_value ≥ ${}_{n}b_{x}$
            else ${}_{n}q_{x}$ ← total_bid_value
    total_bid_value ← total_bid_value − ${}_{n}q_{x}$
```

Offer messages are issued from ${}_{n}q$ notifying the bidding SSP Agents whether their bid has been successful or not, SSP Agents note these offers in their *offers record*, ${}_{m}o$.

This procedure effectively assigns to the SSP Agents that an SCP Agent trusts the most all the supply they want, giving priority to the most trusted Agents first, until all the supply is used up. The remaining Agents are rejected. Other quantification strategies can be implemented, for example equable distribution where each bidder receives a fair share of the supply, but these are not considered further here.

3.3 SSP Utilisation Function

When an SSP Agent has bid for, and received, more units than it actually requires it may return these excess units to unfortunate SCP Agents, who have lost the opportunity to use them and the units are wasted. If demand exceeds offers, the SSP Agent satisfies its customers as best it can, and transfers all the used offers from ${}_{m}o$ to its *utilisation record*, ${}_{m}u$ (full utilisation). When offers exceed demand, ${}_{m}u$ is populated thus:

```
For each SSP Agent m do:
    While total_offer_value > 0
    For SCP Agent x, where x is max(${}_{m}t_{x}$) and ${}_{m}u_{x}$ = 0
    Set ${}_{m}u_{x} \leftarrow {}_{m}o_{x}$ if total_offer_value $\geq {}_{m}o_{x}$
                else ${}_{m}u_{x} \leftarrow$ total_offer_value
    total_offer_value $\leftarrow$ total_offer_value $- {}_{m}u_{x}$
```

The SSP Agents utilises offers from SCP Agents with which it has the best trust relationships preferentially, and risks damaging relationships that are already weaker. Entries in ${}_{m}u$ are transmitted to SCP Agents who made offers as *utilisation messages*, and recorded by the receiving SCP Agent in its *receipts record*, ${}_{n}r$. The SSP Agent suffers no actual penalty, except the loss of credibility with its supplier, for returning offers unused.

3.4 SSP Trust Function

An SSP Agent's trust vector is updated on the basis of the perceived reliability of SCP Agents. This is determined on the basis of whether, or not, an SCP Agent honoured individual bids, ${}_{m}a_{n}$, with corresponding offers, ${}_{m}o_{n}$. A trust function takes two parameters, α ($0 \leq \alpha \leq 1$), the degree to which a positive experience enhances a trust vector element, and β ($0 \leq \beta \leq 1$), the degree to which a negative experience damages the relationship. An individual SSP Agent trust vector element, ${}_{m}t_{n}$, is updated thus:

${}_{m}t_{n} \leftarrow {}_{m}t_{n} - (\beta * {}_{m}t_{n})$, if a bid ${}_{m}a_{n}$ was issued, but no offer ${}_{m}o_{n}$ received, or
${}_{m}t_{n} \leftarrow {}_{m}t_{n} + (\alpha * (1 - {}_{m}t_{n})$, if offer ${}_{m}o_{n} \geq$ bid ${}_{m}a_{n}$ was issued, or
${}_{m}t_{n} \leftarrow {}_{m}t_{n} + ((\alpha * ({}_{m}a_{n} - {}_{m}o_{n})) * (1 - {}_{m}t_{n}))$, if ${}_{m}o_{n} < {}_{m}a_{n}$, or
${}_{m}t_{n}$ is left unchanged otherwise.

These formulations are normalised such that a string of positive experiences asymptotically moves ${}_{m}t_{n}$ towards 1.0, and a string of negative experiences moves it towards 0.0. The function matches our intuition that trust is most enhanced by getting exactly what we requested, partially enhanced by getting some of our request and damaged by being excluded. The formulation also conforms to our expectation that recent experiences are given greater weight that earlier ones, the effect of past events are increasingly discounted with each new experience, but never completely lost. Agents that adopt high values for α are generally more susceptible to single positive experiences, those that adopt a high β value more influenced by negative experiences.

3.5 SCP Trust Function

The SCP trust function is analogous to the SSP trust function, except that it is driven from a comparison of the resource offered, ${}_{n}q_{m}$, against that utilized, ${}_{n}r_{m}$.

${}_{n}t_{m} \leftarrow {}_{n}t_{m} - (\beta * {}_{n}t_{m})$, if an offer ${}_{n}q_{m}$ was made, but no utilisation ${}_{n}r_{m}$ was made,
or
${}_{n}t_{m} \leftarrow {}_{n}t_{m} + (\alpha * (1 - {}_{n}t_{m}))$, if utilisation ${}_{n}r_{m}$ = offer ${}_{n}q_{m}$, or
${}_{n}t_{m} \leftarrow {}_{n}t_{m} + ((\alpha * ({}_{n}q_{m} - {}_{n}r_{m})) * (1 - {}_{n}t_{m}))$, if ${}_{n}r_{m} < {}_{n}q_{m}$, or
${}_{n}t_{m}$ is left unchanged otherwise.

4 Experiments in oTB-Agent Based Trading

We have prepared a simulator in order to investigate the properties of oTB-Agents in the trading situation described previously. The simulation is detailed in that it performs each step in the oTB-Agent algorithm for every Agent at each trading cycle, and emulates every communication message between Agents. The simulator allows the investigator to specify the number of SCP and SSP Agents that will participate, and to set the important parameters for both types of Agent, the α and β trust modification rates; and the overbid rate, the split rate and exploration rate (for SSP Agents). The investigator may single step the simulation, or run it for a pre-determined number of trading cycles, modify parameters and continue. The simulation provides a graphical indication of messages between Agents, and indicates the utilisation of bandwidth resource due to that Agent (as a percentage of the total possible). The investigator may also inspect the trust relationships between any single Agent and its trading partners. At the end of a simulation session logging files may be produced giving a complete record of the development of the trust vectors.

4.1 Experiment One

Experiment one will investigate the effects of load on the relationship between SSP and SCP Agents. We establish trading communities of 10 Suppliers (SCP) and 20

Consumer (SSP) Agents. All SCP Agents are the same (α = β = 0.25), as are all SSP Agents (α = β = 0.25, `srate` = 4, `erate` = 0.2, `obrate` = 0% (i.e. no overbid)). In these experiments all suppliers receive an identical allocation of bandwidth, and all consumers have an equal demand placed on them. All SCP and SSP Agents are identical and treated identically to ensure that the effects of the oTB-Agent procedure are placed in a "fair" trading situation (our first question from section 1).

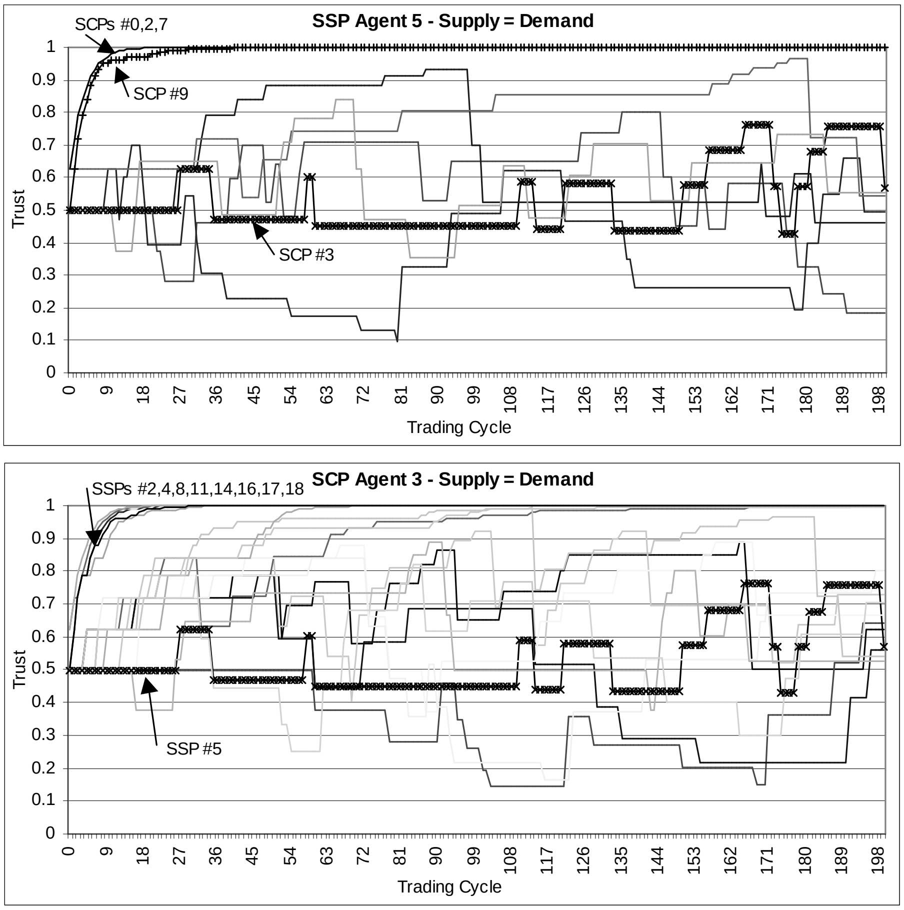

Fig. 3. Trust Relationships Between Agents with Balanced Supply and Demand

This experimental investigation is in three parts, and the results are summarised in figs. 3, 4 and 5. In each part the supply of bandwidth resource is successively restricted in relation to demand, to cause an "overload". Under these circumstances SSPs must develop strong relationships in order to ensure supply (in the converse situation, SCPs are under pressure). Three separate runs are made, one where supply exactly matches demand (100% supply, fig. 3), one where supply is 75% of demand (125% overload, fig. 4) and one where supply is only 50% of demand (150% overload, fig. 5). Each graph in these figures indicates the changing trust relationships of a single Agent (SSP above, SCP below) to all its trading partners. In each case, the SSP graph (top) is one of 20, and the SCP graph (below) one of ten.

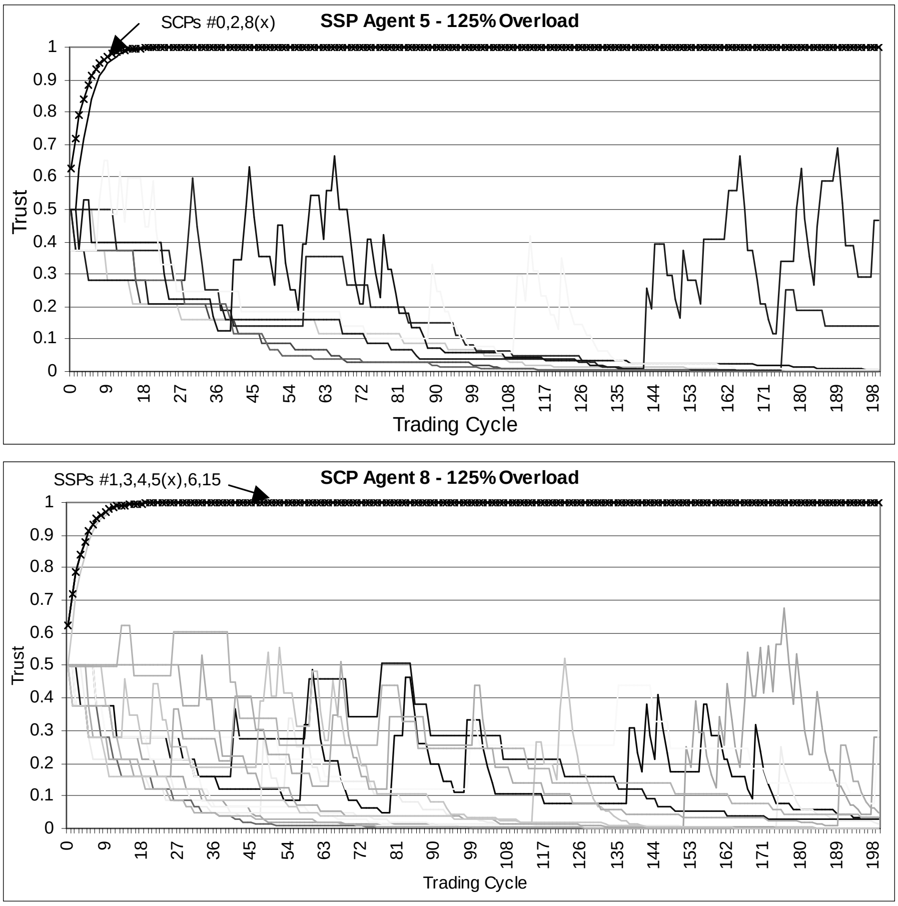

Fig. 4. Trust Relationships Between Agents at 125% Demand Overload

Each figure also highlights the relationship between specific pairs of Agents (fig. 3 between SSP #5 and SCP #3, fig. 4 between SSP #5 and SCP # 8, and fig. 5 between SSP #9 and SCP #3). Note that each run is completely separate, starting with a new random initialisation, therefore Agent numbering in each figure is independent. At the start of each experimental run of 200 trading cycles every trust vector element in all Agents is seeded with an initial value random value in the range 0.499999 and 0.500001. In general, oTB-Agents do not have any “opinion” about the trustworthiness of other Agents (i.e. a trust value of 0.5) at the start of a trading session. This small random perturbation pre-disposes them to start trading with some Agents in preference to others. oTB-Agents are therefore initially *trust neutral*, [15], prior to gaining experience through trading.

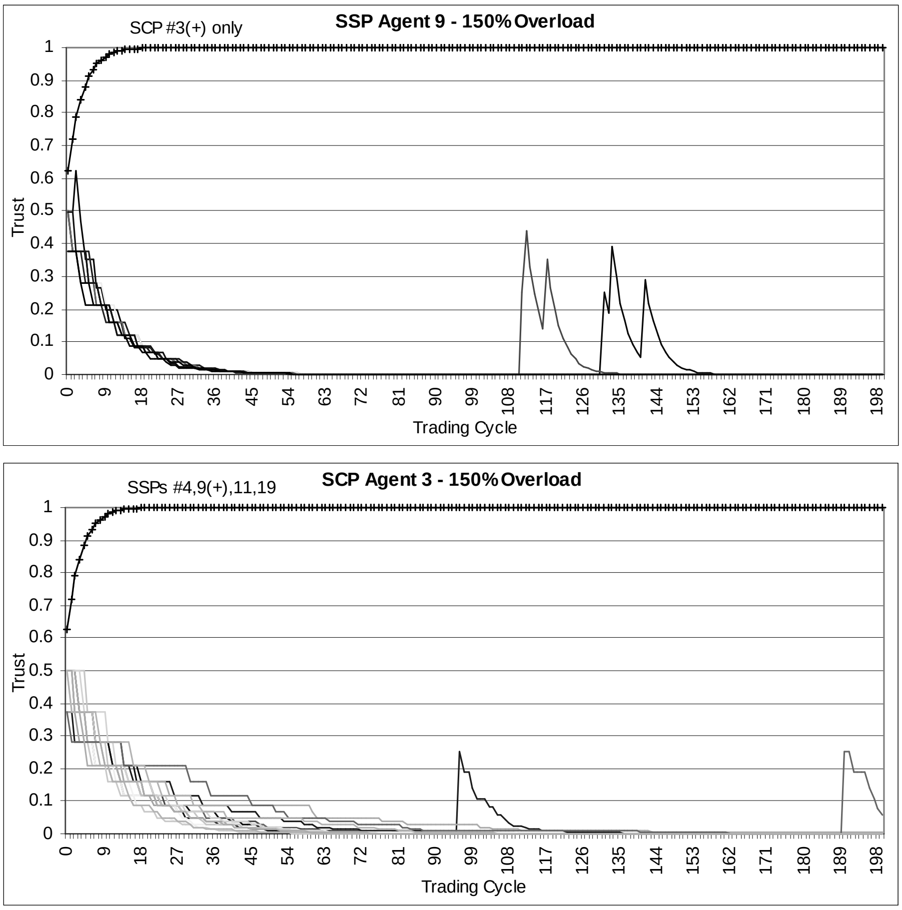

Fig. 5. Trust Relationships Between Agents at 150% Demand Overload

In all instances we see that SSP and SCP Agents tend to “pair-off” very quickly. In the 100% supply case (fig. 3) we can see that SSP Agent builds trust relationships with SCP Agents #0, 2 and 7 quickly, followed by Agent #9 (highlighted with a ‘+’) soon after. These are its preferential trading partners, but it partially trusts many other SCP Agents and trades with them from time to time (this occasional trading between SSP #5 and SCP #3 is highlighted with ‘x’ markers).

In the 125% loading case (fig. 4) we note that this “pairing-off” is more pronounced. Moreover, the number of preferred trading partners has dropped. This indicates that suppliers (who have the upper hand in this situation) prefer to maintain a smaller number of trusted customers, and serve them fully. In turn the customers must continue to bid to these suppliers regularly in order to safeguard their supply of bandwidth. The preferential trading partnership between SSP Agent #5 and SCP Agent #8 is highlighted in the middle row. This effect becomes ever more pronounced as supply is further restricted. At 150% overload (fig. 5) the gulf between those Agents that can trade because they succeeded in establishing a trust partnership and those that did not is very clear. SSP Agent #9 only secured one trust relationship (with SCP #3, highlighted ‘+’), and is clearly going to struggle for supply.

Fig. 6 makes explicit the overall relationship between the degree of trading trust an SSP Agent has been able to secure and its ability to deliver bandwidth to its customers. Each marker in the graph shows the average trust rating for each SSP Agent across all the SCP Agents, against its success in meeting demand. When supply equals demand (100% supply, diamond markers), the overall ability of an SSP Agent to deliver is hardly affected by its perceived trust rating (delivery rate is largely unaffected by overall trust rating). As supply is restricted, (125% overload markers), there is a clear correlation between trust rating and ability to deliver has developed. When supply is further restricted to 50% of demand (150% overload, triangle markers) the correlation is pronounced. The performance of each of the three sub-groups shown circled is directly proportional to the number of suppliers with which the SSP Agent has managed to build a trading relationship. The worst performing group (group 3) only established a partnership with one other SCP Agent (SSP Agents #2, 3, 4, 9 and 13, with an average trust rating of 0.1 and a 0.252% delivery record). The higher group (group 1) comprises Agents #5, 11, 14, 16 and 18, with an average delivery record of 0.729%, established relationships with three suppliers. In this instance no SSP Agent formed a group with four partners under these conditions.

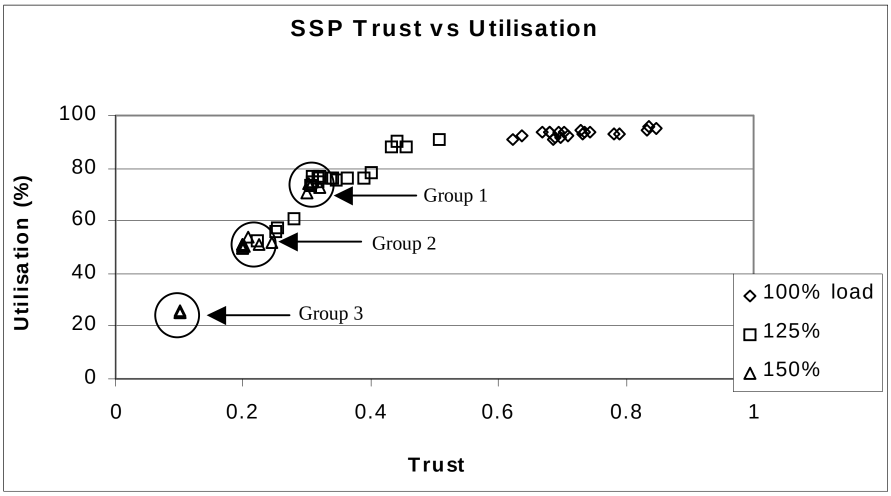

Fig. 6. The Effect of Trust Rating on SSP Delivery Performance

4.2 Experiment Two: The Effects of Changing Circumstances

Experiment two addresses our second question, as to whether establishing a trust relationship over a period of time will equate to loyalty when trading becomes difficult. We repeat the conditions of part one of experiment one (supply = demand), except that at trading cycle 100 supply is reduced to 75% of demand (125% overload). Fig. 7 shows a pair of trust graphs linking the effect on the trust, and hence trade, relationship between SSP Agent #18 and SCP Agent #9 (highlighted ‘x’).

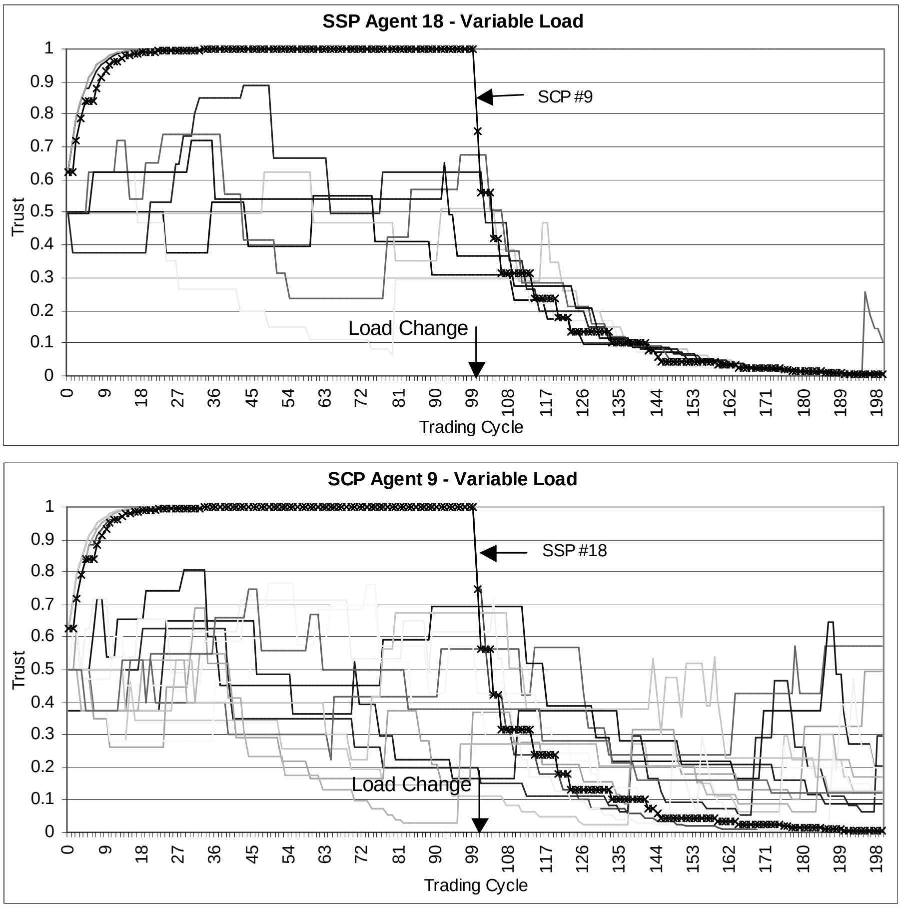

Fig. 7. The effect of Increased Load

It is clear from inspection of these graphs (and the others in the set, not shown), that in addition to the loss of weaker trust relationships (as was the case in experiment one), suppliers have a marked tendency to discard their strong partners on a last-in first-out basis. It appears that, at least in this case, trust does give rise to loyalty.

4.3 Experiment Three: The Effect on Trust of "Greedy" Behaviour

To address our third question, whether an Agent can exploit the "personal" nature of the trust relationship, we perform an experiment in which the SSP Agents are divided equally into two groups. In one group (the "normal" group) they trade "honestly", only bidding for the units of bandwidth they actually require. The second group act "greedily", bidding for 150% of the units they require (`obrate` = +50). Supply is set to equal demand, and the other conditions are as before. Fig. 8 summarises the results obtained from running of this experiment.

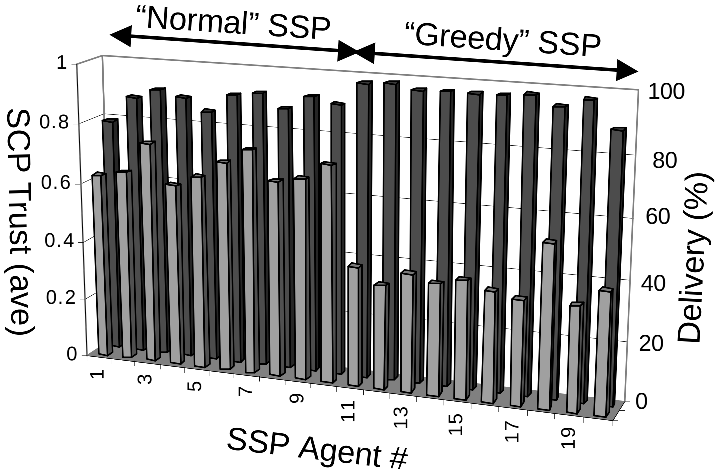

Fig. 8. The Effects of Overbidding

The effects of this greedy behaviour are clear. While the average trust rating by all SCP Agents (front rank) of the greedy SSP Agents is far lower than that for the normal ones (0.391 vs. 0.680), their overall delivery performance (rear rank) is somewhat better (96.1% vs. 88.5%). They perform better because they receive more offers of bandwidth due to overbidding. The effect of the oTB-Agent procedure is to always preferentially buy from your preferred suppliers. So where the normal Agents have good relationships with their preferred suppliers, and reasonable relationships with others, greedy Agents have equally good relationships with their preferred suppliers, but very poor relationships with all the others, who they have treated badly. An element of duplicity, it seems, is still effective in a society where trust is otherwise highly valued.

5 Related Work

There exist a number of issues in the research for security and trust in MAS. We consider three in the context of this work, the role of enforced security measures relative to the social approach, the role of explicit vs. implicit cognitive modelling and the effects of centralised vs. decentralised control in Agent Societies.

5.1 Cryptography and Network Security Techniques vs. Social Approaches

Wong and Sycara, [27] address a number of security and trust issues faced by MAS and provide an infrastructure to deal with such issues. They make use of techniques that are well known in the network security literature, and apply these techniques to MAS. They propose no measures relating to trust or honesty, with no way of ensuring that an Agent will carry out a task as expected, or of guiding an Agent to interact with other Agents that will *probably* be honest. Other approaches of dealing with security issues in MAS have been made by Thirunavukkarasu, Finin and Mayfield, [24] and He, Sycara and Su [12]. These approaches introduced, among a number of things, a number of new KQML performatives enabling Agents to interact in a secure manner. These researchers use classic network security techniques and do not propose any trust models.

It is obvious from this kind of research that network security alone is not sufficient, considering the requirements of multi-agent systems. Following a social approach to security in MAS, Biswas, Debnath and Sen [4] have proposed a model where Agents have relatively complex behaviours. They use a probabilistic mechanism in which an Agent A will decide whether or not to honour a request for help by Agent B. This mechanism takes into consideration previous observations of Agent B, as well as the additional costs incurred by Agent A from Agent B. These researchers demonstrate that Agents that adapt their trust models over time and use probabilistic decision mechanisms are able to successfully withstand the invasion of selfish and exploitative Agents.

oTB-Agents use similar mechanisms for adaptation and decision making. However, unlike oTB-Agents, the Agents of the model of Biswas, Debnath and Sen consider all of their previous observations equally before delegating a task. oTB-Agents, using the trust function described in section 3, place extra weight on recent experiences, although they are influenced by all experiences between the two Agents.

5.2 Implicit vs. Explicit Cognitive Approaches

In implicit approaches, Agents use assessed probabilities to model the trustworthiness of the others. Schillo and Funk [22] conducted a number of simulations where Agents interact with each other using a modification of the prisoner’s dilemma (i.e. the disclosed prisoner’s dilemma with partner selection). Each single Agent builds a model of trustworthiness of the other Agents by gathering data on past behaviour and evaluating averages. When Agents are asked about their knowledge on other Agents, they are free to lie about their observations. Nevertheless, Schillo and Funk show that by averaging the values of a sufficient number of observations Agents can learn models almost twice as fast as other Agents that use only their own observations, while still reaching the same or better accuracy.

Schillo and Funk’s Agents are characterised along two dimensions, being honest/dishonest and altruistic/egoistic. As with [4] the age of an observation is not taken into account. Furthermore no consideration is given as to the reliability of the sources that provide information about Agents.

Castelfranchi, Conte and Paolucci [6] have stressed the importance of reputation in relation to the modelling of trust. These researchers performed a set of experiments in order to simulate the role of reputation in the re-distribution of the costs of norm compliance in agent societies that included normative and non-normative (*cheater*) agents. They showed that communicating knowledge about others' behaviour leads to improved performance of the normative agents. It is important to note that, in contrast to Schillo and Funk's setting, in Castelfranchi, Conte and Paolucci's experiments, the communicating agents did not lie about their observations.

oTB-Agents do not exchange information about their past observations. Clearly, if oTB-Agents communicated their observations (as in the case of Schillo and Funk and Castelfranchi *et al.*) then the 'greedy' Agents of the third experiment would not perform as well as they did. Another major difference between the oTB-Agents and the work that was presented in this section is that, unlike oTB-Agents, the *recency* of an observation is not taken into account.

Explicit cognitive approaches (e.g. [3], [14]) appear more sophisticated, as they attempt to model the "mind" of the other Agents. Castelfranchi and Falcone [7] give a number of guidelines that should be taken into consideration when modelling the trustworthiness of other Agents. These authors separate the concept of trust from that of delegation and mention a number of beliefs that should exist before delegating a task to another Agent (i.e. competence, disposition, dependence beliefs, etc.) Jones and Firozabadi [14], use tools from modal logic to characterise aspects of the reasoning of the Agent who trusts the reliability of the information communicated to it.

5.3 Centralised vs. Decentralised Control Over the Groups of Agents

In organisations where there is a form of centralised control, trust can be viewed as a three party relationship [7]. Agents trust the ability of the authority to assess contract violations and to punish the violators. Agents also trust that other Agents will not violate contracts because they respect/fear the authority. On the other hand, there exist groups of Agents with no form of centralised control. In these groups, Agents need to develop their social skills in order to avoid being exploited by deceitful Agents. The oTB-Agents described here exist in an environment without centralised control.

6 Discussion

The experiments show that oTB-Agents tend to form strong, tight, clusters of trading partners very quickly, and that these partnerships become increasingly important as supply and demand for the traded commodity becomes mismatched. Trust builds trust, but unreliability breeds indifference, "*trust is a peculiar resource which is increased, rather than depleted, through use*" [10]. The Agents modelled here show a clear preference for building strong relationships with trusted partners, sustaining successful partnerships and discarding less trusted partners when conditions turn unfavourable. "Deceitful" Agents, those who generally behave in an untrustworthy manner can still thrive in this community, as long as they maintain good trust relationships with a few key partners.

Reciprocal behaviours in a variety of forms are recognised as effective strategies for forming stable groupings with in larger community [4]. The oTB-Agents defined here appear to adopt an extended "tit-for-tat" attitude, as might be encountered in various game theoretic approaches, such as the Iterated Prisoner's Dilemma ([2], [22], [26]). The scenario presented here differs from the well-understood iterated prisoner's dilemma in that the selection is made on the basis of every transaction between the Agents. Simple Tit-for-tat strategies are considered to be insufficient for most domains of practical interest [4], these problems are largely overcome when the complete transaction history is considered. In addition, SSP oTB-Agents partially base their partner choice on the basis of exploration.

Variations of the formulation used here to evaluate trust find wide application in some of the more numerically orientated approaches to machine learning (such as reinforcement learning, [25], for example) and is ubiquitous, though by no means universal, in theories of natural learning. Jonker and Treur [15] propose a similar formulation for the "quantitative" component of their formalisation of trust. Despite its apparent simplicity the application of this formulation invariably imparts interesting behaviours to the systems that incorporate it.

It is clear from the experiments that a successful first transaction is central to establishing the inter-Agent trading relationship, and an area where the prior assessment of possible partners is critically important. Equally, were this trading community to be augmented with a "reputation" mechanism (such as those of [4], [22] and [28]), by which Agents entering the market could consult existing traders, then the "greedy" Agents of experiment three would be put at a disadvantage. Similar results, i.e. putting "greedy" Agents at a disadvantage, would be obtained in the case where members of the trading community had the ability to *observe* the behaviour and the interactions of the other Agents. In such a case, Agents would have an additional source of information that would enhance their decision-making process concerning their potential trading partners.

7 Current and Future Work

Our aim is to produce a formal specification of the trust model of oTB-Agents that encompasses all three components of a trust based trading relationship, reputation, belief based trust and objective (direct experience) trust. Each, we believe, has an important role to play at different times in the overall life of a trading partnership.

There exist a number of attempts to formalise the concept of trust in MAS ([8], [14], [15]). According to some of these approaches, the formal specification of trust models should include, among other things, decisions such as the use and formation of a trust evolution or update function as well as the properties that should hold for that function [15].

In order to be able to claim that our model of trust is widely applicable, a number of issues are to be addressed in the future. One such issue is the exchanging of observations about other Agents. Consulting other Agents has proved to be helpful in many experiments ([4], [22]). Nevertheless, in the case where a kind of reputation mechanism is used, then the reliability of the Agents information sources should be taken into consideration. Experiential evidence, such as that obtained by oTB-Agents can provide the foundation on which these mechanisms may be built. However it has been shown that the question of what information an agent should accept is a non-trivial task [5]. Several ways have been proposed to order incoming information and consequently decide what information to accept or reject [5]. As Agent behaviour becomes more complex, (e.g. having more opportunities to cheat during interactions), so the modelling of the trustworthiness of other Agents becomes increasingly complex. We intend to integrate our trust modeling with belief analysis and revision [7]. An Agent should be able to evaluate the competence, the willingness and the trustworthiness of another Agent before delegating a task. New parameters should be introduced in the trust modeling process, such as a risk threshold (i.e. how much is an Agent willing to risk the delegation of a task).

Our oTB-Agents trade and develop trust along a single dimension only. More sophisticated Agents will engage with other Agents for a variety of different reasons, and trust should be, in part, a function of the task being performed (Agent X may be reliable when performing task1, but unreliable on task2). We would therefore expect an Agent to maintain an estimate of trust about each task under these circumstances.

A number of other issues are worth considering in future experiments and simulations of Agent communities, such as silent communication, feelings and affective trust. Having to reply to every request can be costly considering communication overheads. A form of silent communication can be adopted in our trading environment, enabling Agents to either refuse to reply to a request or just indicate that they cannot satisfy that request.

Currently, we are working on a formal framework for the specification and logical animation of heterogeneous computational societies. In this framework we define society rules, social roles, social relationships, communication language semantics and social structure. Given such an account of the agent society we will be able to provide a more formal and richer account of the trust modeling of each agent. A multi-agent test-bed [1] is currently being implemented to accommodate our experiments in this formal setting. This test-bed enables the simulation of heterogeneous and possibly antagonistic multi-agent societies and provides a representation of the formally defined social environment of the simulated (trading) communities.

Acknowledgements

This work was partially sponsored by the EC-funded ACTS project AC333 MARINER and partially by the EC-funded ALFEBIITE project (IST-1999-10298).

References

1. Artikis A., Kamara L. and Pitt J. (2001) “Towards an Open Society Model and Animation”, in the Proceedings of Agent-Based Simulation II Workshop, Passau, pp. 48-55
2. Axelrod, R. (1990) “The Evolution of Cooperation”, London: Penguin Books

3. Barber, K.S. and Kim, J. (2000) “Belief Revision Process Based on Trust: Agents Evaluating Reputation of Information Sources”, Workshop on “Deception, Fraud and Trust in Agent Societies”, Autonomous Agents, 2000, pp. 15-26
4. Biswas, A., Debnath, S. and Sen, S. (1999) “Believing Others: Pros and Cons”, Proc. IJCAI-99 Workshop “Agents Learning About, From and With Other Agents”
5. Cantwell, J. (1998) “Resolving Conflicting Information”, in the Journal of Logic, Language and Information, Vol. 7, No. 2, pp. 191-220
6. Castelfranchi C., Conte R., Paolucci M (1998) “Normative Reputation and the Costs of Compliance”, in Journal of Artificial Societies and Social Simulation, Vol. 1, No. 3, http://jasss.soc.surrey.ac.uk/1/3/3.html
7. Castelfranchi, C. and Falcone, R. (1998) “Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification”, ICMAS-98, pp. 72-79
8. Dignum, F. and Linder, B. (1996) “Modelling Social Agents: Communication as Action”, in “Intelligent Agents III: Agent Theories, Architectures, and Languages”, pp. 205-217
9. FIPA (1997) “FIPA-97 Specification, Part 2: Agent Communication Language”, http://www.fipa.org
10. Gambetta, D. (ed.) (1988) “Trust: making and Breaking Cooperative Relations”, Basil Blackwell, Oxford.
11. Griffiths, N. and Luck, M. (1999) “Cooperative Plan Selection Through Trust”, in Proc. 9th European Workshop on Multi-Agent Systems Engineering (MAAMAW ’99), Springer Verlag, Berlin, pp. 162-174
12. He, Q, Sycara, K. and Su, Z. (1998) “Security Infrastructure for Software Agent Society”, in Trust and Deception in Virtual Societies, eds. Castelfranchi and Hua-Tan, pp. 137-154
13. Jennings, N., *et al.* (1999) “FIPA-compliant Agents for Real-time Control of Intelligent Network Traffic”, Computer Networks, Vol. 31, pp. 2017-2036
14. Jones, A. and Firozabadi, B. (1998) “On the Characterisation of a Trusting Agent-Aspects of a Formal Approach”, in Castelfranchi, C. and Hua-Tan (eds.) “Trust and Deception in Virtual Societies”, pp. 155-166
15. Jonker, C.M. and Treur, J. (1999) “Formal Analysis of Models for the Dynamics of Trust Based on Experiences”, in Proc. 9th European Workshop on Multi-Agent Systems Engineering (MAAMAW ’99), Springer Verlag, Berlin, pp. 221 – 232
16. Lorenz, E.H. (1988) “Neither Friends nor Strangers: Informal Networks of Subcontracting in French Industry”, in: [10], pp. 194-209
17. Marsh, S. (1994) “Trust in Distributed Artificial Intelligence”, in Proc. 4th European Workshop on Multi-Agent Systems Engineering (MAAMAW ’92), Springer Verlag, Berlin, pp. 94-113
18. Patel, A., Prouskas, K., Barria, J. and Pitt, J. (2000) “IN Load Control using a Competitive Market-based Multi-Agent System”, in Proc. Intelligence and Services in Networks 2000 (IS&N-2000), pp. 239-254
19. Pham, X.H. and Betts, R. (1994) “Congestion Control in Intelligent Networks”, Computer Networks and ISDN Systems, Vol. 26, No. 5, pp. 511-524
20. Prouskas, K., Patel, A., Pitt, J. and Barria, J. (2000) “A Multi-Agent System for Intelligent Network Load Control Using a Market-based Approach”, in Proc. 4th Int. Conf. on MultiAgent Systems (ICMAS-2000), pp. 231-238
21. Schaerf, A., Shoham, Y. and Tennenholtz, M. (1995) “Adaptive Load Balancing: A Study in Multi-Agent Learning”, Journal of Artificial Intelligence Research, Vol. 2, pp. 475-500
22. Schillo, M. and Funk, P. (1999) “Learning from and about other Agents in Terms of Social Metaphors”, in Proc. “Agents Learning about, from and with other Agents” Workshop.
23. Sholtz, F. and Hanrahan, H. (1999) “Market Based Control of SCP Congestion in Intelligent Networks”, South African Telecommunications Networks and Applications Conference (SATNAC-99)
24. Thirunavukkarasu, C., Finin, T. and Mayfield, J. (1995) “Secret Agents -- A Security Architecture for the KQML”, Proc. ACM CIKM Intelligent Information Agents Workshop, Baltimore, December 1995

25. Watkins, C.J.C.H. (1989) “Learning from Delayed Rewards”, King’s College, Cambridge University (Ph.D. thesis)
26. Williams, B. (1988) “Formal Structures and Social Reality”, in: [10], pp. 3-13
27. Wong, H.C. and Sycara, K. (1999) “Adding Security and Trust to Multi-Agent Systems”, in: Proc. of Autonomous Agents '99 Workshop on Deception, Fraud, and Trustin Agent Societies, pp. 149-161
28. Zacharia, G. (1999) “Trust Management Through Reputation Mechanisms”, Workshop on “Deception, Fraud and Trust in Agent Societies”, Autonomous Agents, 1999, pp. 163-167