Peer Pressure as a Driver of Adaptation in Agent Societies

Hugo Carr[1], Jeremy Pitt[1], Alexander Artikis[21]

[1] Electrical & Electronic Engineering Department,
Imperial College London, SW72AZ
[2] Institute of Informatics and Telecommunications,
National Centre for Scientific Research "Demokritos", Athens 15310

Abstract. We consider a resource access control scenario in an open multi-agent system. We specify a mutable set of rules to determine how resource allocation is decided, and minimally assume agent behaviour with respect to these rules is either selfish or responsible. We then study how a combination of learning, reputation, and voting can be used, in the absence of any centralised enforcement mechanism, to ensure that it is more preferable to behave responsibly than selfishly. This result indicates how it is possible to leverage local adaptation with respect to a set of rules to achieve an intended 'global' system property.

1 Introduction

We are interested in engineering multi-agent systems for applications which require that the system be:

- open: agents are heterogeneous, may be competing, and may have conflicting goals;
- fault-tolerant: agents may not conform to the system specification, but the system should maintain operation, and demonstrate autonomic recovery;
- volatile-tolerant: agents may join and leave the system, but the 'system' itself remains recognisably the same even if all the components change;
- accountable: who performed which action, and to what effect, is significant, so social relations like trust, reputation, responsibility, liability and sanction are all significant;
- decentralised: there is no central mechanism for either knowledge or control, no agent is guaranteed to have full knowledge of the entire system or control over the behaviour of all other components;
- ruled by law: there is a theoretical limit on those making decisions affecting the constraints and/or requirements of behaviour of other components;
- mutable: there is a mechanism by which the specification itself can be changed by the expressed consent of the participants.

Our approach to satisfying these requirements is based on organised adaptation of agent societies. By an agent society we mean a formal specification of:

(1) a set of social constraints (physical capabilities, institutional powers, norms (permissions, obligations, and prohibitions), sanctions, and enforcement policies); (2) a communication language; (3) social structure (roles and the relationships between roles); and (4) other socio-cognitive relations between agents (e.g., in particular, trust). By organised adaptation we mean the intentional modification of such a specification to achieve a commonly-understood goal. This requires understanding (1) what can be adapted (for example, the set of social constraints, or individual behaviour wrt. to that set); (2) when to adapt; (3) how to adapt (e.g. by voting); and (4) evaluating the outcomes of adaptation.

This is a wide-ranging programme of research, but within this paper we focus attention primarily on the interplay of social constraints and relations with respect to the adaptation of individual behaviour to address the issue of fault tolerance (as here understood).

We start from a scenario with multiple agents providing/consuming resources to/from a central repository. However, the set of resources requested is more than those available for distribution, so we define a set of social constraints which determine which agent is allocated resources. Depending on how 'sociably' the agents act during this negotiation, the system can be destroyed, either by agents becoming dissatisfied and leaving the system, or by the over-consumption of resources. However, due to property of decentralisation and autonomy, the incentive to behave 'properly' must come from the agents themselves.

In this scenario, the allocation of resources is decided by a vote. However, Arrow's Impossibility Theorem [1] states that any non-dictatorial voting method can be manipulated by agents expressing a false set of preferences. Voters are therefore are capable of either responsible or selfish behaviour, and have the option to choose between the two. In a system with no social constraints, it is likely that they will objectively find that selfish behaviour yields a higher return.

This suppression of collaboration has been widely studied in game theory as the Prisoner's Dilemma, but can generally be avoided if agents' reputations affect their global social standing [2]. In this paper, we show how the combination of an election to determine the outcome of a negotiation, a reputation mechanism based on voting histories, and a learning algorithm for adaptation of individual behaviour, can be used to ensure that, in the absence of any central enforcement system, it is more preferable (in the long run) to comply with a set of social constraints than it is to violate them.

In the next section, we describe the basic scenario and multi-agent system in more detail. In Section 3 we describe the three primary mechanisms used in this paper: the Q-learning method [3], the reputation mechanism, and election protocol. Section 4 describes experimental results from a 'society' of fifteen agents implemented in the PreSAGE platform [4]. We discuss some related work and draw some conclusions in Section 5. In particular, we note that just by making the assumption of responsible or selfish behaviour (i.e. without comprising the assumption of heterogeneity), individual learning algorithms can be used to optimise outcomes of both individuals and of the society to which they belong.

2 Background

2.1 Scenario and Multi-Agent System

The scenario is based on a 'tragedy of the commons' situation based on the scenario presented in [5]. We also present the animation/simulation platform which we have used to implement the system: further details of the platform can be found in [4].

There is a set of agents U, interacting during a sequence of infinite time slices $t_0, t_1, \ldots, t_n, \ldots$; with each agent requiring, at each time slice, access to resources stored in a bank B.

At each time slice, an agent may be present or absent: the set of agents present at any t is denoted by A_t, $A_t \subseteq U$. To satisfy each of their individual goals, each agent $a \in A_t$ offers, at each time slice, an allocation of resources O_t^a for B, and requests, at each time, an allocation of resources R_t^a from B. We stipulate that, for all $a \in A_t$, $R_t^a > O_t^a$: in other words, the agents can only satisfy each of their individual goals by mutual sharing their collective resources.

Clearly, not all of the requests can be satisfied without 'bankrupting' the system. Therefore, at each t, the set of present agents A_t take a vote on who should have their resource request satisfied. If an agent a receives a number of votes greater than or equal to a threshold τ_t then its request is granted. The problem then is that:

- If τ is too low, too many resources will be distributed, which this will result in the "Tragedy of the Commons" as the system is bankrupted;
- If τ is too high, too few resources will be distributed, which this will result in "voting with their feet" as dissatisfied agents leave the system.

The challenge then is for the agents to agree – again by a vote – a new value for τ in time slice $t + 1$ based on their prediction of how many agents will be present, available resources, and so on. In other words, they are adapting the rule (informally, for a formal expression of the normative rule, see [6, 7]):

the resource controller is obliged to grant access to the resource to a requester, if the number of votes for the requester is greater than or equal to τ

by manipulating the value of τ. We define responsible behaviour to be recognised as voting for an a value of next-τ which will not bankrupt the system or under distribute resources. Consistently voting for a low value of next-τ however, is considered selfish, especially when communal resources are low.

Formally, the external state of the multi-agent system $\mathcal{M}$ is specified, at a time-slice t, by:

$$\mathcal{M}_t = \langle U, \langle A, \rho, B, \mathbf{f}, \tau \rangle_t \rangle$$

where:

$$
\begin{aligned}
U &= \text{the set of agents} \\
A_t &\subseteq U, \text{ the set of } \textit{present} \text{ agents at } t \\
\rho_t &: U \rightarrow \{0,1\}, \text{the presence function s.t. } \rho_t(a) = 1 \leftrightarrow a \in A_t \\
B_t &: \mathbb{Z}, \text{the 'bank', indicating the overall system resources available} \\
\tau_t &: \mathbb{N}, \text{the threshold number of votes to be allocated resources} \\
\mathbf{f}_t &: A_t \rightarrow \mathbb{N}_0
\end{aligned}
$$

The resource allocation function $\mathbf{f}_t$ is constructed by:

$$
\begin{aligned}
\mathbf{f}_t(a) &= R_t^a, \mathbf{card}(\{b | b \in A_t \wedge \mathbf{v}_t^b(\ldots) = a\}) \geq \tau_t \\
&= 0, \text{otherwise}
\end{aligned}
$$

where $\mathbf{v}_t^b : (\ldots) \rightarrow A_t$ is the expressed preference (vote) of agent b in time-slice t, whose inputs are local parameters (in particular agents' reputations) and whose output is a preference array of agents in A_t. Voting for oneself as the most preferred agent is also considered selfish behaviour.

2.2 Simulation/Animation Platform

To animate this system and experiment with different agent behaviours, we have used the agent society animation/simulation platform PreSAGE [4]. PreSAGE is a rapid prototyping tool whose emphasis is on the simulation of agent societies and the social relationships between agents, intended to facilitate the study of the social behaviour of components, the evolution of network structures, and the adaptation of conventional rules. To develop a prototype, it is necessary to define agent participant types: this can be done by extending the abstract class supplied with PreSAGE (to guarantee compatibility with the simulation calls and provide core functionality like message handling etc.) or by defining a new class.

To define the participant class for our purposes, we extend the PreSAGE abstract participant with the following data and functions (we drop the superscript a since it is implicit from context):

$\langle$ Name	a,
Presence	$p : t \rightarrow \{0, 1\}$,
Resources offered	$O : t \rightarrow \mathbb{N}$,
Resources required	$R : t \rightarrow \mathbb{N}$,
π	a set of predictor functions which compute $\lvert A_{t+1} \rvert$
$Q(state, action)$	State-Action evaluations for successful actions to effect **reinforcement Learning**
Reputation Monitor	$r : U \rightarrow \{0, 1\}$
Satisfaction	$\sigma \in [0..1]$,
Satisfaction Increase Rate	$\alpha \in [0..1]$,
Satisfaction Decrease Rate	$\beta \in [0..1]$,
$\mathbf{v}$	voting function which maps a list of agents' historical actions, to an ordered list of agents A_p representing a preference array $\rangle$

The combination of $Q(state, action)$ ie. Q-Learning, reputation tracking and voting are defined in section 3 as the tools we use to show how we can harness peer pressure in the system.

Each time slice sees each active agent follow the system cycle:

Phase 1: set threshold
  $A_t = \{a | a \in U \wedge \rho_t(a) = 1\}$
  each agent $a \in A_t$ uses π^a to propose and **vote** on a value for τ_t
Phase 2: resource request
  each agent $a \in A_t$ offers resources O_t^a, and requests resources R_t^a
  each agent $a \in A_t$ computes **reputation** values for each agent $b \in A_t$
Phase 3: resource assignment
  each agent $a \in A_t$ uses $\mathbf{v}^a$ to **vote** for a vector of agents comprised of $a \in A_t$
  $\mathbf{f}_t$ is computed from the votes cast and τ_t
Phase 4: update
  B_t is updated according to the resources allocated
  each agent updates its satisfaction rating (see below)
  each agent updates its Q-Value estimates for **reinforcement learning**

Phase 4 is where an agent evaluates its personal success and potentially changes its behaviour to try to improve its standing. In section 3, we elaborate in more detail how our participants use their Q-Values and reputation monitors towards this goal.

2.3 Related Research

Although the use of learning techniques to change system parameters is addressed in [8, 9], the scenario described here defines an institution to be the sum of its participants rather than a separate entity. The research most closely aligned with the current work is the foundational work of Axelrod [10] on the

evolution of norms[1]. In this work, he posited a *norms game*, in which an individual has an opportunity to defect against a norm, as determined by its propensity to *boldness*. If it defects, then it gets a positive payoff, and all the others get a negative one. A defecting agent may or may not be seen (to defect); if it is seen by another agent then the agent may choose to punish or not, as determined by its propensity to *vengefulness*. If it chooses to punish then the punished agent (the defector) gets a large negative outcome, and the punishing agent a small negative outcome (i.e. there is a cost associated with enforcing a norm).

Axelrod ran computer simulations where the population changes over a sequence of generations, whereby those agents with more successful boldness/vengefulness strategies produce more descendants than less successful ones (keeping a fixed population size). The outcome was, starting from an average level of boldness and vengefulness: first boldness fell, because it was costly to be bold when vengeance was (relatively high); then vengefulness fell, as was is costly to be vengeful without direct benefit; then boldness rose sharply, destroying the restraint originally shown – as Axelrod notes: "a sad but stable state" ([10]:p1100).

To redress this situation, Axelrod introduced a variant of the game with a *metanorm*, in this case the punishment of defection may or may not be seen, and not punishing itself may be punished. So there is some incentive to be vengeful. Axelrod simulations now showed that if a population started with 'sufficient' vengefulness the restraint could be maintained, but if not, then the metanorms game collapsed just like the norms game.

To some extent, the scenario in this paper is a partial reconstruction of Axelrod's norms game, with some important variations. The four phases of each time slice are comparable to one path in the norms game, where (Phase 1) the agents vote either selfishly or responsibly (defect or do not defect); (Phase 2) every agent sees what each other agent has done; (Phase 3) agents punish defectors through the reputation mechanism; and (Phase 4) agents update behaviour through Q-learning (notionally equivalent to the production of the next generation).

However, in our scenario, there is pre-established conventional rule, and the norm is to effect a 'sociable' adaptation of that rule. Therefore we do not deal with generations of agents and the evolution of norms, but with one generation (whose numbers may change, e.g. when a new selfish agent is introduced) and the robustness of its norms wrt. maintaining a stable state in face of potentially disruptive components (i.e. when a new selfish agent is introduced). Our agents do not perform game-theoretic decision making along boldness vs. vengefulness dimensions, but instead express a preference (a vote) based on a larger number of local parameters, thus the internal complexity of our agents, while hidden from general view, is greater than the individual players of Axelrod's game.

[1] N.B. An Axelrod norm refers to a general social norm, which can arguably be deconstructed in terms of the norms defined by Pitt et al [7]. As an example, a guest at a dinner party is generally obliged to request permission to smoke from an empowered entity; which by convention is the host. We do not make a distinction between defecting against an Axelrod norm, or a Pitt norm.

Furthermore, the two votes required, one for the value of τ and one for the candidate order, require each agent to express preferences. This signalling introduces an element of communication which side-effects the game, and in combination with the reputation system and learning (individual adaptation) provides the robustness to resist the disruption of selfish agents.

3 Mechanisms for Peer Pressure

3.1 Overview of mechanisms

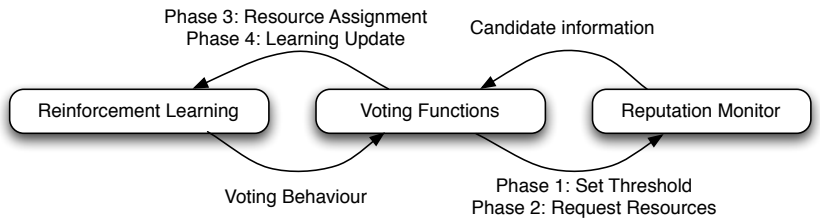

Fig. 1. The mechanisms and dataflow for each timecyle

Figure 1 illustrates the cycle through which we can maintain a stable system with self-enforcing behaviour using peer pressure. The voting functions and patterns of each agent are public, and will feed into the reputation monitors of each participant. The reputation monitor then generates a list of preferred agents derived from how socially each agent is perceived to be acting. This vote generates a result which depending on a win or loss of resources, drives the learning algorithm of an agent. An agent will then choose a voting behaviour (social or antisocial) which the learning algorithm calculates to give the highest return.

3.2 Voting Functions

As defined in section 2.1, τ will symbolise the threshold number of votes an agent requires from the population to receive resources. In order to select an appropriate value we must ensure that the electorate is not split. To avoid this we select τ using a two round election system where round one consists of suggestions for τ, and round two a vote between the two most popular suggestions. This way even if the responsible voters are split on their suggestions, it is probable that at least one of the popular choices is closer to theirs than a selfish one.

Once τ has been decided we can move on to resource requests, offers, and the main vote for who will receive resources this time slice. For simplicity we have fixed the resource requests and offers as we believe the significance of the introduction of a normal distribution would not justify the increase in complexity.

We have found a plurality vote in this round to be ineffective for discouraging antisocial behaviour. Statistically the ideal value of τ for a system cycle, tends to be less than or equal to the number of votes that an agent is granted to use. For example if each agent is allowed to vote for two candidates to receive resources, the value of τ which will ensure a stable resource stockpile for a responsible population, tends to oscillate around two. Therefore if we allow agents to use these votes for themselves, selfish behaviour will almost always be rewarded. We need to force agents to be less introspective, as the solution to this problem lies in the opinion an agent has of its neighbours.

To this end we have changed the system's main voting protocol to Borda which requires agents to vote in the form of an ordered preference list[2]. Repetitions in the list will be ignored and incomplete lists will be penalised. Agents behaving selfishly are loyal to no one and will rank themselves before anyone else. Responsible agents however, will conscientiously rank the voters, leaving selfish agents with no extra Borda points. It will therefore be the agents behaving responsibly who will receive on average a larger number of Borda points. The interpretation of a Borda preference array can be viewed in table 1 in which x = the number of agents receiving points; we define this to be approximately half of the total population.

Preference Array	Vote weight
p_1	x
p_2	x-1
.	.
.	.
p_{x+1}	2
p_x	1
p_{x-1}	0
.	.
.	.
p_m	0

Table 1. Borda preference array interpretation used when aggregating the votes

3.3 Reputations and Behaviours

For responsible behaviour in Borda elections, agents must diligently rank their neighbours in a preference relation. This requires the use of a reputation monitor which can distinguish between selfish and social behaviour. Then depending on individual reputation values the preference relation can be constructed. For

[2] τ in this case, represents the minimum number of Borda points required to receive resources.

agents which have identical reputations we make sure to randomise their positions between one another. This can be achieved by basing a preference relation on a random variable which takes the reputation as an input rather than a strict ordering based on historical behaviour.

Due to their activity profiles agents do not have perfect knowledge of the system. They may rejoin with no reputation information on a number of new agents. It is important that we carefully define how to recognise antisocial behaviour as quickly as possible. The system has therefore been specified in terms of two poles of behaviour, responsible and selfish. Agents behaving responsibly are defined to be altruistic capitalists, putting the needs of the system before their own, but expecting some sort of return for their efforts. The priority lies in avoiding the tragedy of the commons and bankrupting the system. Responsible agents also have a duty to the system to exclude agents which behave selfishly.

Agents behaving selfishly are somewhat simpler than their responsible counterparts as their duty is only to themselves. They will always vote so that they are first on their preference list and try for the lowest value of τ available. These actions cannot be hidden from their neighbours so the gamble is that the 'pro' of voting for oneself will offset the 'con' of a poor social standing. Needless to say, a system comprised solely of selfish agents would bankrupt, so our aim is to create a set of norms that rewards responsible behaviour while punishing selfish.

The difficulty is finding a method that predicts a τ value responsibly, but avoids a universal consensus. In the El-Farol Bar problem a group of entities with access to the same information and the same prediction facilities will unanimously decide on an action. The example cited in [11] describes a population of residents who use the same function to predict if it is worth going to the local bar. If the bar is between 50% and 60% capacity, then the patrons will have a good evening, otherwise the bar will either be too full or too empty. However if all agents come to the same prediction, the bar will always either be completely full, or completely empty.

The solution lies in a range of prediction functions randomly initialised and distributed among the agents. Agents will use their predictor functions with historical information of what a good value of τ would have been in the previous timecycles and verify this with what the best value would have been for the current time cycle. Functions returning a value close to this will be ranked above less accurate predictions. This should result in an easily observable 'responsible' behaviour when compared to agents who always vote for $\tau = 0$.

We employ a set of predictors, each constructed using a randomly weighted average of historical values. We take x_i to be a random value between zero and one adhering to a uniform distribution.

$$w_i = \frac{x_i}{\sum_{\forall j} x_j}$$

$$pred = \sum_{\forall i} w_i.a_i$$

where a_i refers to the historical values.

The historical information is selected by collating all the votes and ranking the most popular agents. We then hypothetically give each agent resources until all the offered resources have been exhausted. The last agent to receive resources then becomes the benchmark for τ, and we enter the number of votes that it received as the historical τ threshold.

3.4 Reinforcement Learning

Reinforcement learning is a non-essential addition to the experiment but is useful to demonstrate how an initially selfish agent can be 'rehabilitated' through peer pressure. Q-Learning specifically provides us with an unbiased evaluation of sets of actions. For example, if a selfish agent is successfully excluded from the agent society will receive no resources causing the $Q(state, action)$ value for selfish behaviour to continually decrease. An agent becomes 'fed up' with the status quo when the $Q(state, selfish)$ value falls below that of the $Q(state, responsible)$ value. It is at this point that an agent will try behaving responsibly to see if this provides a better return.

In accordance with Axelrod's work on cooperation [2] the rest of the agents will quickly forgive a repentant agent and cease ostracising it; as manifested by a better average position in the preference array vote of a responsible agent. This will in turn increase the number of resources allocated to an agent and therefore raise the $Q(state, responsible)$ value. This illustrates the intuitive interpretation of a $Q(state, action)$ value: a metric representing the previous success of an action in a system state.

Q-Learning can be based on a Markov Decision Process which takes into account the history of success and failures of actions in a state transition system. However due to agent activity profiles and an agent's incomplete knowledge of the system, participants were occasionally prone to rapid switching of behaviours. In an attempt to remedy this, we introduced the threshold suggested in [3] resulting in Delayed Q-Learning; we used this in conjunction with a Q-learning rate suggested in [12].

We define the system in terms of actions $x \in X$ from states $s \in S$, with buffers of size m saving reward information r_k at time k. The state-action value function can then be defined using:

$$Q_{t+1}(s, x) = \frac{1}{m} \sum_{i=1}^{m} (r_{k_i} + \gamma V_{k_i}(s_{k_i})) + \epsilon$$

where

$$\begin{aligned} V_t &= max_{x \in X} Q_t(s, x) \\ r_k &\in [0, 1] \\ \gamma &\in [0, 1] \end{aligned}$$

In our system we have only two Q-Values to update as we have only one state and two 'actions' representing responsible and selfish behaviour.

Each state-action pair has its own buffer allowing it to evaluate all of the available actions for a state. But to ensure a delay, the algorithm only updates state-action Q-values once the buffer has been filled, after which it is then reset. New Q-Values must change the original by more than 2ϵ or they will be rejected. This gives agents longer to evaluate state-action pairs

Learning takes place during the declaration of the election result. The dominant behaviour will interpret the list of winners as $r \in \{0, 1\}$, and depending on how full the buffer is etc. will update the Q-value for this action accordingly. Behaviour for a timecycle is always selected using

$$a' := argmax_{a \in A} Q_t(s, a)$$

4 Experimental Results

4.1 Agent Animation

Agents begin their life cycle with a role assignment we assume to have been established in advance. This can be done through a role assignment protocol as outlined in [7]. The chair of the session then calls for participation in the system, and the voters send confirmation messages. A voter has an activity profile which is linked to a Markov chain, resulting in a stable population, but as mentioned in previous sections they may refuse to participate if they no longer consider the system to be viable. A confirmation of participation is tantamount to a commitment to provide resources in this time slice, regardless of the result.

During the τ selection vote, agents' votes are kept public so they can update their reputation monitors accordingly. Specifically an agent is disapproved of if they vote for an unreasonably low value of τ. We define this to be a vote for $\tau = 0$ when the stockpile of resources leftover from the previous timeslice is less than zero ie. the system is in debt. This makes the recognition categorical as a responsible agent would not perform this action. This allows us to focus on whether the reputation monitor works in the absence of flawed information.

The main vote follows from the τ selection by opening a ballot for the agents who voted in the first round. It is at this point that agents need to form their preference arrays and send them to the chair for collation. After a defined time-out the chair will accept no more votes and calculate how many resources are left in the system post distribution. For brevity the optimal value of τ, for use in the prediction functions, is calculated by the chair and circulated with the election results. This reduces the complexity of the system significantly.

Once agents know whether they have received resources this timeslice, they are able to update how successful their state-action pair was this time slice, and adjust their satisfaction rating. Satisfaction is representative of an agent's overall success in the system and is maintained parallel to individual action histories. We define satisfaction to lie between zero and one, and to be governed by the equations:

$$\sigma_{t+1} = \sigma_t + (1 - \sigma_t)\,\alpha$$
$$\sigma_{t+1} = \sigma_t - \sigma_t \beta$$

where the former is used to improve an agent's satisfaction in a society, and the latter to regress it. α and β represent the satisfaction increase and decrease rates respectively.

4.2 Experiments

Initially we show that this experiment is stable amongst a group of these agents who have already established a responsible moral context. We do this by setting the initial Q-Values of responsible behaviour higher than the selfish. We then add a potentially destabilising element to the system at timecycle 3000, taking the form of a set of agents whose selfish Q-values are higher than their responsible ones. We will examine how and whether the learning algorithm works in conjunction with the reputation monitor. We should observe a change in behaviour matching a delayed change in satisfaction as the social standing of an agent improves. If an agent doesn't manage to learn, they should have a very low satisfaction by the end of the simulation.

4.3 Results

We include here an example of a simulation of 15 agents. We start with a community of 10 responsible and stable agents, and introduce a set of 5 selfish agents. In this example four of the initially selfish agents learn to behave responsibly, but one does not. We have chosen a small example as it is difficult to demonstrate trends in this system. Trends of the selfish population exist only insofar as they either change their behaviour and get absorbed into the responsible population, or they don't and leave the system. These two outcomes can happen at any time and statistical analysis tends to average away all the interesting details[3].

Figures 2 and 3 show the average satisfaction of the initial population of responsible voters to be stable. Even when the selfish agents are introduced at cycle 3000 the satisfaction seems to remain high universally. What is actually happening here however, becomes clearer on looking at figures 4 and 5.

Figure 4 shows that most of the selfish agents are learning to change their quite rapidly, resulting in a better reputation, and a higher satisfaction. However on comparing figures 3 and 5 we see that as the system gradually recognises agent 13 as a selfish element, the 'Benefit of the Doubt' eventually disappears, and it receives no more resources, resulting in an extremely low satisfaction rating. It is effectively ostracised and ends up leaving the system altogether.

4.4 Summary of Results

The experiments reported here offer additional supporting evidence for Axelrod's original claims, make their own contribution, and serve as a basis for a successively richer set of experiments in further work. The experiments confirm,

[3] We have performed this experiment several times with fifty agents, and found the system to behave in the same way.

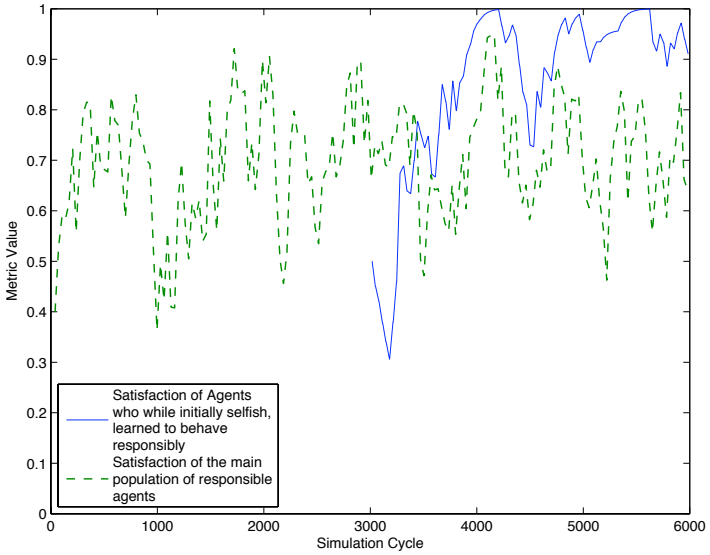

Fig. 2. Satisfaction of Selfish and Responsible Agents

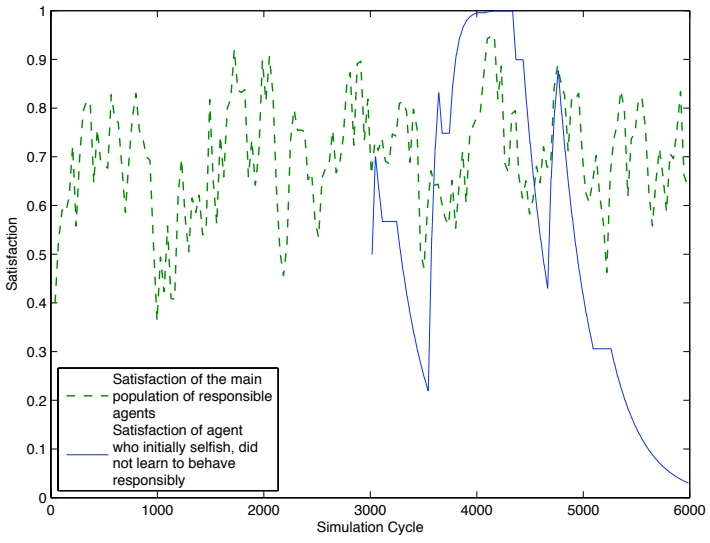

Fig. 3. Satisfaction of Responsible Agents and Agent13

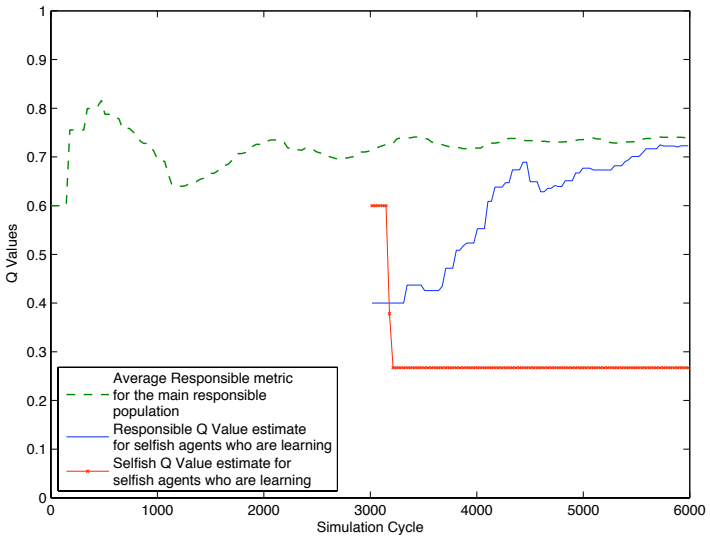

Fig. 4. Q Value estimates of the Responsible Agents and the initially Selfish Agents

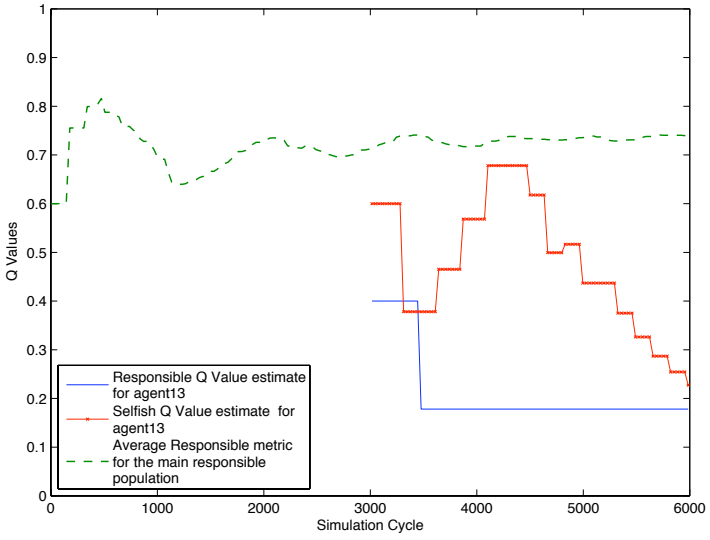

Fig. 5. Q Value estimates of the Responsible Agents and Agent13

as stated by Axelrod, that norms and conventions are a powerful mechanism for resolving conflicts of interest in disputes between multiple parties even in the absence of a central authority. In addition, social norms (e.g. the 'norm' is to vote for a 'reasonable' value of τ) and social constraints (i.e. the reputation mechanism) work well in preventing minor defections given that the cost of enforcement is low. In their own right, the experiments show how effective it is to give control over the adaptation of rules to those whose outcomes are most directly affected by the adaptation (cf. [13]), and how it is possible to leverage local adaptation with respect to a set of rules to achieve an intended 'global' system property.

There are several lines of further investigation opened up by this work. One is a more fine-grained behaviour rather than responsible or selfish. Rather, we would have a propensity to selfish behaviour, and correspondingly allow a propensity to punish. This would necessitate a more subtle implementation of *forgiveness* which is an important element of autonomic systems for self-repair [14]. A second line of investigation concerns a peer to peer system allowing 'gossiping' between agents to allow groups to converge their opinions. For this, we could use the models of opinion formation formalised by [15].

5 Summary and Conclusion

In this paper we have outlined an agent society which maintains fault tolerance through peer pressure. We chose three mechanisms to create this dynamic: The reputation monitor, the reinforcement learning, and the voting functions out of which emerged emerged a group of responsible agents which when pitted against selfish individuals effectively pressurised the latter into their preferred way of behaving (i.e conforming to a norm, in the sense of Axelrod). If agents refused to conform they were eventually permanently identified as anti-social and received no votes at all.

Through the platform PreSage, we have shown that a social norm can be enforced in a system with a strong moral pretext without the use of a centralised enforcement agency, given that the cost of enforcement is low or non-existent. However, it would be interesting to investigate the effects of scale (size of population) and the corresponding increased cost of a more centralised enforcement mechanism. For example, in a relatively small society, enforcement could be based on peer-pressure, word-of-mouth or other reputation mechanism (as here) with low or no cost. In a relatively large society, central reputation registers could be provided, with punishment provided by the equivalent of a 'police force', but at a much higher cost.

On a closing note, we also agree with Axelrod [10] when he observes that the probabilistic effects and complexities of population diversity make it difficult (if not impossible) to determine the consequences of a given behavioural model. However, computer simulation techniques offer a viable alternative which can reveal the system dynamics and stable states, as well as specific influence of identified agent behaviour profiles.

6 Acknowledgements

The first author is supported by a UK EPSRC studentship. The second author is supported by the EU FP6 Project ALIS 027958. Thanks for support to Lloyd Kamara, Brendan Neville, and Daniel Ramirez-Cano

References

1. Arrow, K.J.: A difficulty in the concept of social welfare. Journal of Political Economy **58** (1950) 328
2. Axelrod, R.: The Evolution of Cooperation. (2006)
3. Strehl, A.L., Li, L., Wiewiora, E., Langford, J., Littman, M.L.: Pac model-free reinforcement learning. In: ICML '06: Proceedings of the 23rd international conference on Machine learning, New York, NY, USA, ACM (2006) 881–888
4. Neville, B., Pitt, J.: Presage: A programming environment for the simulation of agent societies. Proceedings AAMAS Workshop on Programming Multi-Agent Systems (PROMAS) 2008
5. Carr, H., Pitt, J.: Adaptation of voting rules in agent societies. Proceedings AAMAS Workshop on Organised Adaptation in Multi-Agent Systems (OAMAS) 2008
6. Artikis, A., Kamara, L., Pitt, J., Sergot, M.: A protocol for resource sharing in norm-governed ad hoc networks. In: Proceedings of Workshop on Declarative Agent Languages and Technologies (DALT). Volume LNCS 3476. Springer (2005) 221–238
7. Pitt, J., Kamara, L., Sergot, M., Artikis, A.: Voting in multi-agent systems. Comput. J. **49**(2) (2006) 156–170
8. Bou, E., Lopez-Sanchez, M., Rodrguez-Aguilar, J.A.: Towards self-configuration in autonomic electronic institutions. To appear In Coordination, Organization, Institutions and Norms in agent systems, Lecture Notes in Computer Science. Springer Verlag, 2007 (Jun 2007) 16
9. Bou, E., Lopez-Sanchez, M., Rodrguez-Aguilar, J.A.: Using case-based reasoning in autonomic electronic institutions. To appear in proceedings of the COIN@Durham'07 (Aug 2007) 14
10. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review **80**(4) (1986) 1095–1111
11. Arthur, W.: Inductive reasoning and bounded rationality: The el farol problem (1994)
12. Even-Dar, E., Mansour, Y.: Learning rates for q-learning. J. Mach. Learn. Res. **5** (2004) 1–25
13. Ostrom, E.: Governing The Commons: The Evolution of Institutions for Collective Action. Cambridge University Press (1990)
14. Vasalou, A., Pitt, J.: Reinventing forgiveness: A formal investigation of moral facilitation. In: iTrust. (2005) 146–160
15. Ramirez-Cano, D., Pitt, J.: Follow the leader: Profiling agents in an opinion formation model of dynamic confidence and individual mind-sets. Intelligent Agent Technology, 2006. IAT '06. IEEE/WIC/ACM International Conference on (Dec. 2006) 660–667