Friends No More: Ostracism in Multi-Agent Systems

Adrian Perreau de Pinninck, Carles Sierra, and Marco Schorlemmer

IIIA – Artificial Intelligence Research Institute
CSIC – Spanish National Research Council
Bellaterra (Barcelona), Catalonia, Spain
adrianp|sierra|marco@iiia.csic.es

Abstract. In a normative society there are two main problems: defining norms and enforcing them. Enforcement becomes a complex issue as societies become more decentralized and open. We propose a distributed mechanism to enforce norms by ostracizing agents that do not abide by them. The simulations have shown that, although complete ostracism is not always possible, the mechanism reduces substantially the number of norm violation victims.

1 Introduction

In an open multi-agent system (MAS), there is no easy control over who is allowed to join the system. Open MAS are composed of autonomous agents of all types, without a predefined structure. In a normative MAS a set of norms are added to restrict the set of available actions in order to improve the coordination between them. An autonomous agent has the choice whether or not to support a norm. It is up to the agent to decide if it is convenient for it to abide by it. If following a norm is more profitable, it is in the agent's own interest to act as the norm establishes. But this is not always the case, as some norms are only profitable when all agents abide by them. For this kind of norm, an agent that does not adhere (i.e, a violator) will profit at the expense of the agents that adhere. We call this type of agent a free-rider.

A real life example of an open MAS is Gnutella.[1] Gnutella is a pure peer-to-peer (P2P) file sharing application without centralized servers. Peers can share files from their hard drives so that others can download them. Each peer contains a list of other peers (i.e friends or neighbors) with which it can interact. A peer can carry out two actions: search for peers that have a file it wants, and download the file from any of them. Peers are found by asking neighboring peers if they or any of their neighbors have the file. This process is recursive. Once a list of peers that share the file are returned to the querying peer, it can choose one of them from which to download the file. Anyone with a Gnutella compliant software can join the MAS, as long as it has a list of other peers with which to interact.

[1] http://www.gnutella.org

Although this system works, users can use the network to download without sharing any files themselves. Therefore, the burden of sharing is carried only by some of them, while the rest benefit from the service. In order to solve this problem the following norm could be added: "Users must share files in order to download". But since the norm is not built into the Gnutella protocol, Gnutella peers can choose whether or not to adhere to it. A mechanism is needed so that norms can be enforced in a P2P network such as Gnutella.

The purpose of this paper is to present a distributed mechanism that attains norm compliance by ostracizing norm violating agents in an open MAS. Our scenario allows agents to interact with each other. An agent can interact with the agents it is linked to directly or indirectly through a linked path (i.e. agents can interact with direct neighbors, or with neighbors of neighbors, and their neighbors and so on...). An initiator agent will search for a path in the society to find a partner agent with which to interact. All the agents in the path that are not the initiator or the partner agent will be called mediator agents (i.e, agents mediating the interaction).

We use a game-theoretic approach to interactions. They are modeled as a two-player game with two possible strategies; cooperate and defect. The utility function will be that of a prisoner's dilemma (see Figure 1), since the total utility gained by both players is maximized if both players cooperate, and the maximum utility to be gained by a single agent is maximized if it defects while the other cooperates.

PD	Cooperate	Defect
Cooperate	3,3	0,5
Defect	5,0	1,1

Fig. 1. Prisoner's Dilemma Payoff Matrix

The norm in this scenario is for agents to cooperate with each other, thus attaining the maximum utility for the society. Nonetheless, agents can choose to ignore the norm and defect (i.e., violate the norm). Violators are better off because they prey on norm-abiding agents and gain more utility. In order to attain norm enforcement, some agents (we will call them enforcer agents) are given the ability to stop interacting with violators, and to stop them from interacting with the enforcer's own neighbors. When enough agents use this ability against a violator, it will be ostracized.

The ostracism process can be seen in Figure 2. At first an undetected violator in the network (the dark gray node) can interact with all the other agents (light gray nodes are liable to interact with the violator). When the violator interacts, it can be detected by enforcer agents which will start blocking its interactions (black nodes are blocking agents, and white nodes are agents that the violator cannot interact with). When all the violator's neighbors block it, it is ostracized.

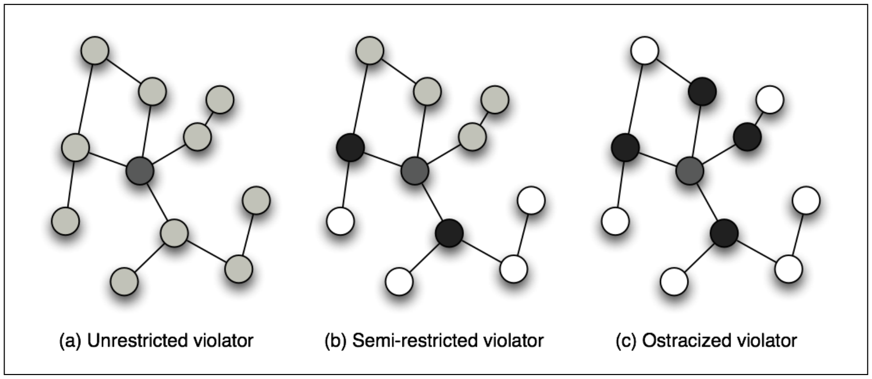

Fig. 2. Ostracizing a violator

Gossip is essential to find out information about other agents in a distributed environment. We will use gossip as part of the enforcement strategy in order to ostracize agents. Since we want gossip to take up as little resources as possible, gossip information is given only to the agents mediating the interaction. If agent ag_v violates the norm when interacting with agent ag_1, ag_1 may spread this information to all mediator agents so they may block ag_v in the future.

By running a set of simulations, we study under which conditions the mechanism works, and give measures of its success (such as the violations received or the utility gained). Our hypothesis are: (1) Norm violations can be reduced by applying a simple local blocking rule. (2) The way agents are organized influences the enforcement capabilities. (3) The enforcement strategy used by enforcer agents can reduce the number of violations received by norm-abiding agents which do not enforce norms.

In Section 2 we describe related work in the area of norm enforcement. In Section 3 we present a detailed description of the scenario in which the simulations will be run. In Section 4 we describe the simulations and we analyze the resulting data. In Section 5 we present the future work that will follow from this research.

2 Related Work

The problem of norm enforcement is not new. It has been dealt with in human societies (also an open MAS) through the study of law, philosophy, and the social sciences. Recently it is being dealt with in computer science, specially since norms are being studied as a coordination mechanism for multi-agent systems. Axelrod [1] first dealt with the application of norms from an evolutionary perspective. Enforcement is seen by Axelrod as a sort of meta norm to punish agents that do not punish violators. The norm game is often modeled as an N-Player Iterated Prisoner's Dilemma [1, 8]. In these cases the norm is to cooperate and ways are sought to ensure agents prefer cooperation. Other research studies see norms as a way to avoid aggression or theft [4, 7, 11, 13]. In these cases agents gain utility by either finding items or receiving them as gifts. But these items can be stolen by another agent through aggression, which is why possession norms are added that avoid aggression.

Mainly, two main enforcement strategies have been studied in order to attain norm compliance: the use of power to change the utilities through sanctions or rewards [2, 3, 8, 12], and the spread of normative reputation in order to avoid interaction with violators [4, 6, 7, 11, 13]. In both cases researchers have tried to find ways to make norm adopters better off than norm violators. But this is not always accomplished [4, 7].

Norm enforcement models have been suggested in [2, 6]. They show how violating the norm becomes an irrational strategy when punishment is possible. But these models assume the following: (1) That agents are able to monitor other agents' activities; (2) and that agents have the ability to influence the resulting utility of interactions. Assumption (1) can be materialized in two ways; by having a central agent mediate all interactions [2], or by having agents recognize violators through direct interaction with them, or through gossip with other agents. The first solution does not scale, since the mediator agent would be overwhelmed with information in a large system. The second scales, but it is less efficient. Assumption (2) can be carried out through third-party enforcement [2], or self-enforcement. Using a third party does not scale [6] because the third party can easily be overwhelmed. For self-enforcement, all agents must have the ability to affect the outcome utility of interactions.

Axelrod [1] proposes the "shadow of the future" as a reasonable mechanism to affect an agent's choice in the iterated prisoner's dilemma game. An agent is deterred from defecting because the probability of interacting with the same agent in the future is high, and agents will defect in future interactions with known violators. Nonetheless, this method does not impose sanctions, since the ability to enforce material sanctions is contradicts the agent's inherent autonomy [6]; no technique has been given to impose utilitarian sanctions. How can a sanction be applied to an agent who refuses to pay? A possible method is the threat of ostracism or physical constraint. Conte and Castelfranchi have studied the possibility of avoiding interaction with norm-violators, but this is not the only factor in ostracism. Ostracism means excluding someone from the society, which implies not just avoiding interaction with the ostracized agent but also preventing it from interacting with anyone.

Kittock [9] was the first to study how the structure of a multi agent system affected the emergence of a social norm. He studied regular graphs, hierarchies, and trees. In [5] Delgado studied emergence in complex graphs such as scale-free and small-world, and in [10] studied the relationship between a graph's clustering and emergence.

3 The Scenario

We model our multi-agent system as a undirected irreflexive graph. $MAS = \langle Ag, Rel \rangle$, with Ag the set of vertices and Rel the set of edges. Each vertex models an agent and each edge between two vertices denotes that the agents are linked to each other. Different structures for an agent society are possible. We have chosen three for their significance: Tree, Random, and Small-World. We define a tree as a graph in which each node has one parent and some number of children; one node, the root node, has no parent. Nodes are linked to both their parents and children. A random graph is without structure, any node can be linked to any other one with a given probability. A small-world graph is created by starting with a regular graph, and adding a small number of random edges.

We use a game-theoretic approach by modeling interactions as a two player prisoner's dilemma game. The norm is that agents ought to cooperate (i.e. an agent disobeys the norm by defecting). In order for two agents to interact, there must be a path in the graph between the two (i.e. only neigbors, and neighbors of neighbors can interact). One agent will search for a path that leads to another agent with which to interact. The searching agent we will call *initiator agent*, the agent chosen to interact we will call *partner agent*, and the rest of the agents in the path we will call *mediator agents*. The partner finding process is explained below, but first we need to formally describe some terms.

We define the set of neighbors of an agent a_i as the set of agents it is linked to directly in the graph: $Neighbors(a_i) = \{a_j \in Ag \mid (a_i, a_j) \in Rel\}$. Each agent also has a set of agents it blocks (an agent cannot block itself): $Blocked(a_i) \subseteq Ag \setminus \{a_i\}$. An agent a_i can query another agent a_j about its neighbors. We denote the set of agents that a_j answers with $reportedNeighbors(a_i, a_j) \subseteq Neighbors(a_j)$. This set depends on the blocking strategy of a_j. The diferent strategies will be explained below. A path is a finite (ordered) sequence of agents $p = [a_1, a_2, \ldots, a_n]$ such that for all i with $1 \leq i \leq n-1$ we have that $a_{i+1} \in Neighbors(a_i)$, and for all i, j with $1 \leq i, j \leq n$ and $i \neq j$ we have that $a_i \neq a_j$. The agent a_1 of a path is the initiator agent, agent a_n is the partner agent, the rest are mediator agents.

In order to find a partner, the initiator agent a_i creates a path $p = [a_i]$ with itself as the only agent in it. The initiator agent will then query the last agent in the path (the first time it will be itself) to give it a list of its neighbors. It will choose one of them[2] (a_j) and adds it to the end of the path $p = [a_i, ..., a_j]$. At this point, if agent a_j allows it, the initiator agent can choose agent a_j as the partner. Otherwise, it can query agent a_j and continue searching for a partner.[3]

Once the partner is chosen a prisoner's dilemma game is played between the initiator and the partner. The game results and path are given to each of the playing agents. Playing agents can choose to send this information to

[2] To avoid loops, an agent that is already part of the path cannot be chosen again.

[3] It may happen that a path's last element is an agent that refuses to play a game with the initiator agent, and will return an empty list of agents when queried for its neighbors. In that case backtracking is applied.

all the mediators in the path. This is what we call gossip, which contains the agents' names and their strategy choices for the given game: $Gossip = \langle ag_i, choice_i, ag_j, choice_j \rangle$, where $choice_i$ and $choice_j$ are either *cooperate* or *defect*.

During the whole process agents can execute any of the following actions:

– Return a list of neighboring agents when asked for its neighbors.
– Choose one of the agents of a list as a mediator.
– Choose an agent as partner for an interaction.
– Choose a strategy to play in the PD game when interacting.
– Inform mediators of the outcome of the interaction.

Our society of agents will be composed of three types of agents, each one having different strategies for the actions it can execute. The *meek agent* is the norm-abiding agent that always cooperates. It will always return all its neighbors to any agent that asks, it will choose an agent randomly from the list as the mediator, with probability p it will choose the mediator as the partner and with probability $1 - p$ will ask it for its neighbors, it will always cooperate in the PD game, and finally it will do nothing independently of the game outcome. The *violator agent* has exactly the same strategies as a meek agent, except that it will always defect when playing a game.

Finally, the *enforcer agent* is the one with the ability to block violators, which is essential in order to achieve their ostracism. An enforcer agent has the same strategies as the meek agent with the following exceptions: It will add agents that have defected against it to the set of blocked agents, and will inform all mediators when this happens. If an enforcer is informed of the results of a game it was mediating, it will act as if it had played the game itself. Enforcer agents will never choose an agent in their blocked set as a partner, and will not allow an agent in their blocked set to choose it as a partner. Therefore, a violator agent will never be able to interact with an enforcer who is blocking it. When an enforcer agent is asked to return a list of its neighbors by an agent who is not in its blocked set, two different strategies are possible. The Uni-Directional Blockage (UDB) strategy, where all its neighbors will be returned ($reportedNeighbors(a_i, a_m) = Neighbors(a_m)$). Or the Bi-Directional Blockage (BDB) strategy, where only those neighbors not in its blocked set are returned ($reportedNeighbors(a_i, a_m) = Neighbors(a_m) \setminus Blocked(a_m)$). When the querying agent is in the enforcer agent's blocked set it always returns an empty set.

4 Simulations

The simulations are going to be run using the scenario specified in section 3. Each simulation consists of a society of 100 agents. The society will go through 1000 rounds, in each round agents will take turns to find a partner with which to interact, one turn per round. If an agent cannot find a partner it skips a turn.

The interaction is modeled as a prisoner's dilemma with the utility function in Figure 1.

The parameters that can be set in each simulation are:

- Percentage of Violators (V) - from 0% to 50% in 10% increments[4].
- Percentage of Enforcers (E) - from 0% to 100% in 10% increments[5].
- Type of Graph (G) - either hierarchy, small world, or random.
- Enforcement Type (ET) - Uni-Directional Blockage (UDB), or Bi-Directional Blockage (BDB).

An exhaustive set of simulations have been run with all the possible values for each parameter. Each simulation has been run 50 times in order to obtain an accurate value. The metrics that have been extracted are: the average number of games played, the mean violations received, and the mean utility gained by an agent. The standard deviation has been calculated for each one of these metrics. The metrics have been calculated for the whole society and for each type of agent.

The data gathered from the simulations support our hypotheses. The graph in Figure 3 shows that the higher the percentage of norm-abiding agents that use a blocking rule the lower the average number of norm violations received by any agent in our system. There are five different lines drawn in the graph, each one stands for a different percentage of violating agents. It is intuitive that the higher the percentage of violator agents, the higher the number of norm violations perceived by any agent in the system. In all cases a higher enforcer to meek agent ratio (x-axes) leads to lower violations received per agent (y-axes). When the ratio of enforcers is high, violators end up interacting with each other. Since the y-axes measures the violations received by "any" agent, the improvement seen in Figure 3 is not significant. The data referring to the violations received only by norm-abiding agents shows a higher improvement.

We also deduce from the data that different organizational structures in the multi-agent system influence the norm enforcement. In Figure 4 we have extracted the average norm violations (y-axes) for each of the different structures tested: Random, Small World, and Tree. We have only used the simulations where violator agents account for 20% of the population, therefore at most there will be an 80% of enforcers. The x-axes contains the different percentages of enforcer agents tested. It can be seen that both random and small world networks have an almost identical graph line. On the other hand the tree structure has shown to improve the enforcement capabilities. As an interesting side note, the tendency is that the more enforcer agents, the less violations. But in random and small world networks, when the percentage of enforcer agents reaches its maximum the percentage of violations received are increased. We believe this happens because in both these networks violator agents manage to find paths

[4] We will not consider societies with more than half the agents being violators since in that case the norm should be to defect.

[5] The percentage of meek agents is computed through the following formula: $M = 100\% - V - E$. Therefore, $V + E$ cannot be more than 100%.

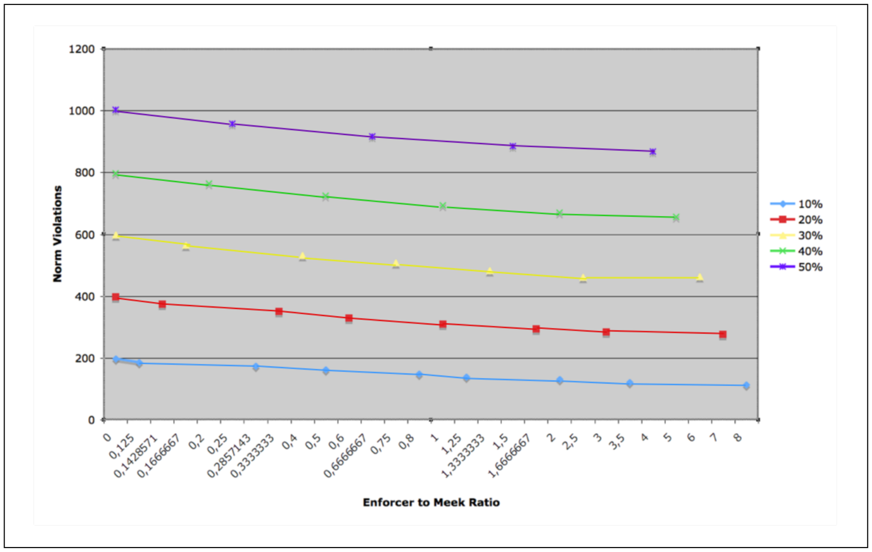

Fig. 3. Use of local blocking rule reduces violations

that link each other. Since at this point there are no meek agents for them to prey on, they are forced to interact with each other. In an interaction between two violator agents, two violations are being accounted for and the average of violations is increased.

The last hypothesis we made in Section 1 was that the enforcement strategy used by enforcer agents impacts the number of violations perceived by meek agents. We have presented the data in Figure 5 that supports this hypothesis. The x-axes shows the enforcer to meek agent ratio. The higher the ratio the more enforcer agents. The y-axes contains a metric for the increment in efficiency at protecting meek agents from violations. The efficiency is calculated by getting the ratio of violations perceived by meek agents (not any agent as in the previous two graphs) for each of the two different enforcement strategies, and calculating the difference in percentage. $\mathit{Efficiency} = ((\mathit{Violations}_{BDB} / \mathit{Violations}_{UDB}) - 1) \times 100\%$. In Figure 5 we observe that for random and small world networks the efficiency is positively correlated with the enforcer to meek agent ratio. We can conclude that Bi-Directional Blockage has a higher efficiency at protecting meek agents from violator agents. This cannot be extended to the tree network. In this case the efficiency stays along the 0% line with some deviations. We argue that in networks organized as trees, the choice of enforcement strategy does not have a significant influence in the outcome. The reason might be that the tree network is already good for ostracizing offenders, and the blockage strategy does not improve on that.

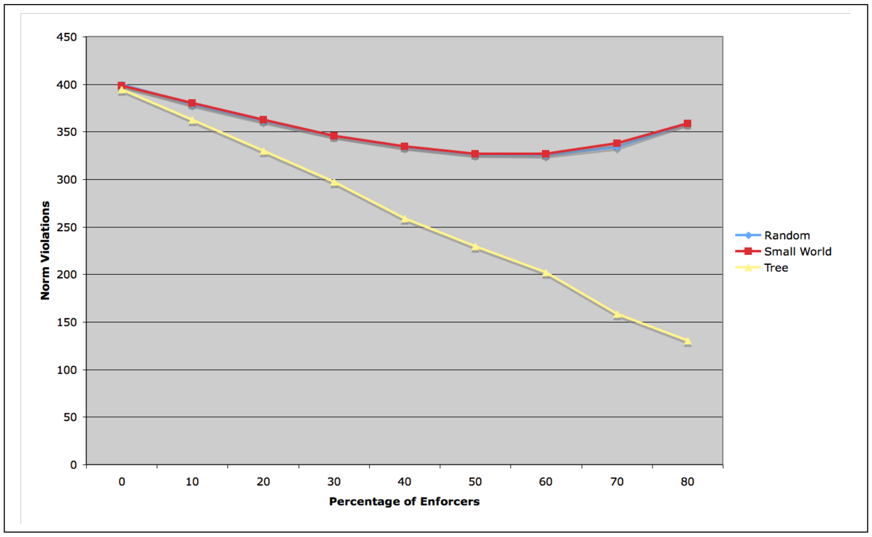

Fig. 4. Enforcement capabilities vary depending on structure

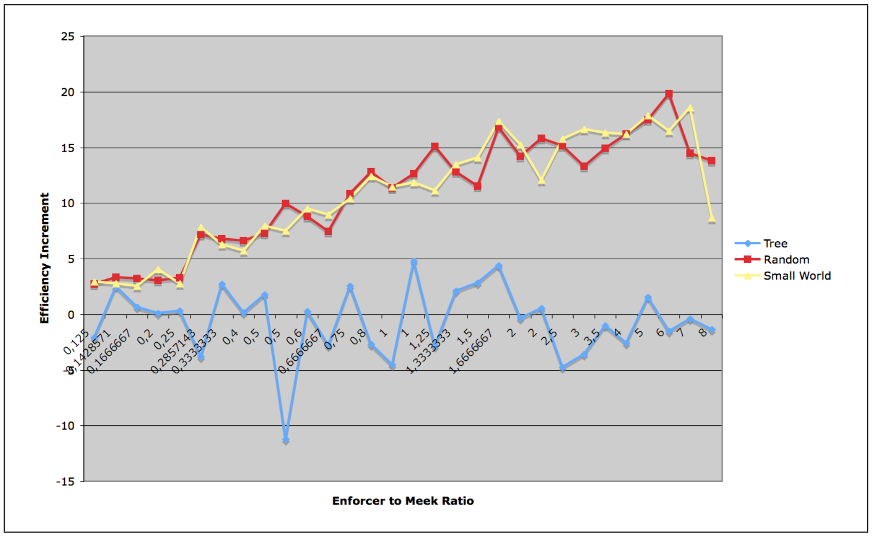

Fig. 5. Enforcement strategy influences received violations

5 Further Work

This paper is part of ongoing research on norm enforcement. The data extracted from the simulations has yet to be deeply analyzed. We think that more information can be extracted from it. We have yet to analyze the impact of blockage on each agent type, this paper has presented information mostly about the impact on any agent. We have to analyze the impact of blockage on the amount of utility gained by the system. When interacting, agents play the prisoner's dilemma, which tends to benefit those who defect. We would like to test whether our approach to ostracism makes cooperating rational from an utilitarian perspective.

We want to analyze the impact of other network parameters (e.g., clustering factor, diameter, number of links per agent). We have seen that a tree network is better from an enforcement perspective, but we want to find out which characteristics of a tree make this possible.

Finally, other studies have shown that the efficiency of enforcement diminishes when enforcement conveys a cost to the enforcing agent [1, 8]. We would like to add cost to the mixture in following scenarios. Also our scenario is completely static, and if we try to model something similar to a real network we need to simulate dynamic networks too.

Acknowledgements

This work is supported under the OpenKnowledge[6] Specific Targeted Research Project (STREP), which is funded by the European Commission under contract number FP6-027253. The OpenKnowledge STREP comprises the Universities of Edinburgh, Southampton, and Trento, the Open University, the Free University of Amsterdam, and the Spanish National Research Council (CSIC).

A. Perreau de Pinninck is supported by a CSIC predoctoral fellowship under the I3P program, which is partially funded by the European Social Fund. M. Schorlemmer is supported by a *Ramón y Cajal* research fellowship from Spain's Ministry of Education and Science, which is also partially funded by the European Social Fund.

References

1. Robert Axelrod. An evolutionary approach to norms. *The American Political Science Review*, 80:1095–1111, 1986.
2. Guido Boella and Leendert W. N. van der Torre. Enforceable social laws. In *AAMAS*, pages 682–689, 2005.
3. Jeffrey Carpenter, Peter Matthews, and Okomboli Ong'ong'a. Why punish: Social reciprocity and the enforcement of prosocial norms. *Journal of Evolutionary Economics*, 14(4):407–429, 2004.

[6] `http://www.openk.org`

4. Cristiano Castelfranchi, Rosaria Conte, and Mario Paoluccci. Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation*, 1(3), 1998.
5. Jordi Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141(1):171–185, 2002.
6. Amandine Grizard, Laurent Vercouter, Tiberiu Stratulat, and Guillaume Muller. A peer-to-peer normative system to achieve social order. In *COIN*, 2006.
7. David Hales. Group reputation supports beneficent norms. *Journal of Artificial Societies and Social Simulation*, 5(4), 2002.
8. Douglas D. Heckathorn. Collective sanctions and compliance norms: a formal theory of group-mediated social control. *American Sociological Review*, 55(3):366–384, 1990.
9. James E. Kittock. The impact of locality and authority on emergent conventions: initial observations. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 420–425, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.
10. Josep M. Pujol, Jordi Delgado, Ramon Sangüesa, and Andreas Flache. The role of clustering on the emergence of efficient social conventions. In *IJCAI*, pages 965–970, 2005.
11. Adam Walker and Michael Wooldridge. Understanding the emergence of conventions in multi-agent systems. In Victor Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems*, pages 384–389, San Francisco, CA, 1995. MIT Press.
12. Fabiola López y López, Michael Luck, and Mark d'Inverno. Constraining autonomy through norms. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 674–681, New York, NY, USA, 2002. ACM Press.
13. Stephen Younger. Reciprocity, sanctions, and the development of mutual obligation in egalitarian societies. *Journal of Artificial Societies and Social Simulation*, 8(2), 2005.