

Sociologically Inspired Engineering



David Hales
University of Bologna
Italy
dave@davidhales.com



Bruce Edmonds
Manchester Metropolitan University
United Kingdom
bruce@cfpm.org

It is always hard to predict the kinds of cross fertilisations that will occur between different areas within the interdisciplinary hot-house of agent-based research. It is now widely established that biologically-inspired mechanisms are applicable as basis for the engineering of distributed and robust systems. For example, many adaptive agent algorithms draw inspiration from biological evolution, and there has been much recent work on ant-inspired routing methods to name just two. We discuss here recent results of cross fertilisation within agent-based research that is less widely known: recent developments by which mechanisms derived from social theories, particularly those expressed computationally through

phisticated) individual behavioural and learning rules and then executing simulations to determine what kinds of societies and structures emerge. The focus of much of this research has been on trying to gain a greater understanding of human societies — the world we actually live in. Interestingly, some of the mechanisms that have been discovered are directly relevant to open engineering issues in distributed systems.

As was described in AgentLink News 14 by Steve Phelps and Peter McBurney [7], ideas from computational economics are informing new automated market design algorithms for deployment in multi-agent systems. Many of these ideas got

...recent developments by which mechanisms derived from social theories, particularly those expressed computationally through agent-based social simulation, are being applied to tough engineering problems in distributed settings.

agent-based social simulation, are being applied to tough engineering problems in distributed settings.

The idea of using “social metaphors” for thinking about self-organising software is not new. Marvin Minsky’s classic A.I. text “Society of Mind” [5] explicitly envisages minds as composed of semi-autonomous entities with coalitions, conflicts and hierarchies. What is new is the application of techniques from multi-agent-based social simulation (MABS) to the hard distributed engineering problems that agent researchers are currently attempting to solve.

For well over a decade, simulation-friendly social scientists have been using multi-agent-based simulations to develop, test and communicate mechanisms of social emergence. This means programming individual agents with (often so-

started when progressive economists, attempting to understand what methods real people use to solve such problems, started making computational models of such methods.

A general problem that pervades much of agent research is that of maximising system-level performance while allowing individual agents reasonable levels of autonomy. In many situations there arises a contradiction between these two aspects. This kind of dilemma happens in human societies all the time, for example, when someone decides not to pay on a short train ride (free-ride) or evade tax by not declaring income. One way to stop these behaviours is to impose draconian measures via centralised government control — ensuring all individuals behave for the common good — so as to stop free riders. However, this is costly and hard to police, as well as it raises other issues such as who polices the police? In the par-

lance of agent-oriented systems: the method does not scale well, is sensitive to noise, and has a high computational overhead.

In the context of actually deployed massively distributed software systems, Peer-to-Peer (P2P) file sharing applications (such as KaZaA and eDonkey) have similar problems — most users only download files rather than sharing them. This limits the effectiveness of such systems. Even when the P2P client software is coded to force some level of sharing, users may modify and redistribute a hacked client. It has been noted that P2P file sharing is one of the applications in which only a small number of altruists are needed to support a large number of free riders. Consequently, it can be argued that this might be why popular P2P applications tend to be limited to sharing only files rather than processor or distributed storage, for example.

This sort of case can be seen as an example of a more ‘fundamental’ issue: how can one maintain cooperative (socially beneficial) interactions within an open system under the assumption of high individual (person, agent, or peer) autonomy. An archetype of this kind of social dilemma has been developed in the form of a minimal game in which two players each selected a move from two alternatives and then the game ends and each player receives a score (or pay-off), the Prisoner’s Dilemma game (see below, if you are not familiar with it).

The Prisoner’s Dilemma

Table 1 shows a so-called “pay-off matrix” for the two-player game. If both cooperate then both get a reward, the score R. If both defect they are punished, the score P. If one player defects and the other cooperates then the defector gets T (the temptation to defect), the other getting S (the sucker). When these pay-offs, which are numbers representing some kind of desirable utility (for example, money), obey the constraints $T > R > P > S$ and $2R > T+S$, then we say the game represents a Prisoner’s Dilemma. When both players cooperate this represents maximising of the collective good but when one player defects and another cooperates this represents a form of free-riding. The defector gaining a higher score (oh, the temptation!) at the expense of the co-operator (who then becomes a sucker, really).

	cooperate	defect
cooperate	R, R	S, T
defect	T, S	P, P

Table 1. The Prisoner’s Dilemma Game: two players select either to cooperate or defect.

A game theoretic analysis of the prisoner's dilemma drawing on the Nash equilibrium solution concept (as defined by the now famous John Nash) captures the intuition that a utility-maximising player would always defect in such games because, whatever the other player does, a higher score is never attained by choosing to cooperate. The Nash Equilibrium might be a partial explanation for why there is so much free-riding on existing P2P file-sharing systems — users are simply behaving to maximise their utility. However, do we have any way to solve this problem without going back to centralised control or closed systems? A Nash analysis gives us a good explanation for selfish behaviour but not for altruistic behaviour. As stated earlier, even in P2P file sharing systems there are some altruists (keeping the show on the road).

It has been argued by many researchers from the social and life sciences that human (and even animal) societies produce much more cooperation than a Nash analysis would predict. Consequently, various cooperation promoting mechanisms (often using the prisoner's dilemma as their test bed) have been proposed over the last half-century. Within the last decade, there has been a flood of new results in the area due to the increasing availability of computational power and the advent of agent-based social simulation.

This work is characterised more by the experimental method than by the development of a *prior* formal theory. This is because the real dis-

tributed systems that are the target of this work are complex and only partially known, forcing theory to be a matter of falsifiable hypothesis and experiment. Thus this work is often closer in method to the natural sciences than to the formal sciences [3].

For well over a decade, simulation-friendly social scientists have been using multi-agent-based simulations to develop, test and communicate mechanisms of social emergence.

We now concentrate on describing two mechanisms, imported from the social simulation literature, that have been directly applied to the problem of maintaining high levels of cooperation in file-sharing P2P applications. One is actually deployed and working; it is called BitTorrent and uses mechanisms popularised by the political scientist Robert Axelrod. The other, offering a potentially more robust and scalable method, applies some very recent ideas from social simulation within a prototype simulation environment.

BitTorrent [2] employs a form of the tit-for-tat strategy popularised in the 1980's by computer simulation tournaments of an iterated version of the prisoner's dilemma. Researchers were asked to submit programs (agents, if you like) that repeatedly played the game against each other [1]. The result of all these tournaments was that a simple strategy called tit-for-tat did remarkably well against the majority of other submitted programs (although other strategies can also survive within the complex ecology that occurs when there is a population of competing strategies).

Tit-for-tat (TFT) operates in environments where the prisoner's dilemma is played repeatedly with the same partners for a number of rounds. The basic strategy is simple: an agent starts by cooperating then, in subsequent rounds, copies the move made in the previous round by its opponent. This means defectors are punished in the future: the strategy relies on future reciprocity. To put it another way, the "shadow" of future interactions motivates cooperative behaviour in the present. In many situations this simple strategy can outperform pure defection.

In the context of BitTorrent, a TFT-like strategy plays a significant role in ensuring cooperation. The basic BitTorrent mechanism is simple: files are split into small chunks (about 1MB each) and downloaded by peers, initially from a single hosting source. Peers then effectively "trade" such chunks with each other using a TFT-like strategy — i.e., if two peers offer each other a required chunk then this equates to mutual cooperation. However, if either does not reciprocate then this is analogous to a defect, and the suckered peer will retaliate in future interactions.

The process is actually a little more subtle because each peer is constantly looking at the upload rate / download rate from each connected peer in time — so it does not work just by file chunk but by time unit within each file chunk. While a file is being downloaded between peers, each peer maintains a rolling average of the download rate from each of the peers it is connected to. It then tries to match its uploading rate accordingly. If a peer determines that another is not uploading fast enough, then it may "choke" (stop uploading) to that other.

Additionally, peers periodically try connecting to new peers randomly by uploading to them — testing for better rates. This means that if a peer does not upload data to other peers (a kind of defecting strategy), then it is punished by those other peers which will not share file chunks with the "defector" in the future — hence a TFT-like strategy based on punishment in future interactions is used.

Axelrod used the TFT result to justify sociological hypotheses — such as understanding how fraternisation broke out between enemies across the trenches of World War I. Cohen has applied a modified form of TFT to produce a file sharing system resistant to free-riding. However, TFT has certain limitations, it requires future interactions with the same individuals, and each has to keep records of the last move made by each opponent. What about situations in which many interactions are with strangers with no promise of future



**16 –17 December 2004
Barcelona, Spain**

The aim of this second European Workshop on
Multi-Agent Systems
is to encourage and support
activity in the research and development
of multi-agent systems,
in academic and industrial
European efforts.

For details, see www.eumas.org/2004

interaction and/or take place between millions of nodes? This is often the case in large-scale file sharing systems and might be true for other kinds of P2P applications such as storage and processor sharing. In such circumstances, TFT-type strategies are of limited use — does the result that in a single-round game it often pays to free-ride mean that such systems are doomed to be taken over by free-riders?

Recent work, drawing on agent-based simulations of cooperative group formation based on “tags” (social labels or cues) and dynamic social networks suggests a new mechanism which does not require reciprocal arrangements. It is based on the idea of cultural group selection and the well-known social psychological phenomena that people tend to favour those believed to be similar to themselves — even when this is based on seemingly arbitrary criteria (e.g. supporting the same football team). Despite the rather complex lineage, like TFT, the mechanism is refreshingly simple. Individuals interact in cliques (subsets of the population). Periodically, if they find another individual who is getting higher utility than themselves they copy them — changing to their clique and adopting their strategy. Also, periodically, individuals form new cliques by joining with a randomly selected other.

Defectors can do well initially, suckering the co-operators in their clique — but ultimately all the co-operators leave the clique for pastures new — leaving the defectors all alone with nobody to free-ride on. Those copying a defector (who does well initially) will also copy their strategy, further reducing the free-riding potential in the clique. So a clique containing any free-riders quickly dissolves but those containing only co-operators grow.

Given an open system of autonomous agents, all cliques will eventually be invaded by a free-rider who will exploit the clique hence dissolving it. However, so long as other new cooperative cliques are being created, then cooperation will persist in the overall population. In the context of social labels, or “tags”, cliques are defined as those individuals sharing particular labels (e.g.,

supporting the same football team). In the context of P2P systems, the clique is defined as all the other peers each peer is connected to (i.e., its neighbourhood) and movement between cliques follows a process of network “re-wiring”.

Through agent-based simulation, the formation and maintenance of high levels of cooperation in the single round prisoner’s dilemma and in a P2P file sharing scenario (containing over

A general problem that pervades much of agent research is that of maximising system-level performance while allowing individual agents reasonable levels of autonomy.

100,000 peers) can be tested. This has been done and these mechanisms have been shown to be effective [4]. The mechanism appears to be highly scalable with zero scaling cost — i.e., it does not take longer to establish cooperation in bigger populations. There are a number of outstanding issues that need to be addressed before this new technique can be deployed. However, as previously mentioned, a deployable system would have applications beyond file sharing.

Here we have outlined just two mechanisms, arising from social science via agent-based social simulation, targeted towards P2P applications. However, there are many other areas in which the same kind of cross-fertilisation is in the process of occurring. These include areas such as trust, reputation, market processes, group formation, normative regulation, negotiation, and even emergent ontologies. For a flavour of recent agent-based social simulation work, take a look at the *Journal of Artificial Societies and Social Simulation (JASSS)* [6], as well as the MABS and ESOA series of workshops (both published in Springer’s LNAI series). A workshop currently being organised will look specifically at the use of metaphors and ideas originating in social phenomena to the engineering of complex computational systems (details below).

Increasingly, software engineers are working with distributed and noisy systems through necessity.

They are tackling, essentially, sociological and economic questions — though many do not realise this. Agent-based simulation offers a large body of algorithmically specified mechanisms and techniques that can go a long way towards solving these problems. In the AgentLink II roadmap, engineering with social metaphors was placed as a medium to long-term possibility. It seems that this is developing more quickly than expected. Increasingly, we may see cutting-edge software engineering and social science intersect, through mutual self-interest, into a single grouping or, dare we say, clique. You will be able to recognise when this has occurred — a clique composed of sociological computer scientists and computer-literate sociologists is likely to be tagged by being fairly scruffy!

References:

1. Axelrod, R. (1984). *The evolution of cooperation*. Basic Books, NY.
2. Cohen, B. (2003). *Incentives Build Robustness in BitTorrent*. Presented at the 1st Workshop on the Economics of Peer-2-Peer Systems, June 5-6, 2003, Berkeley, CA. (Available at: www.sims.berkeley.edu/research/conferences/p2pecon/).
3. Edmonds, B. and Bryson, J. (2004). *The Insufficiency of Formal Design Methods — the necessity of an experimental approach for the understanding and control of complex MAS*. Presented at AAMAS 2004, July, New York. (Available at: <http://www.aamas2004.org>).
4. Hales, D. (2004) *From selfish nodes to cooperative networks - emergent link based incentives in peer-to-peer networks*. Proc. of the 4th IEEE International Conference on Peer-to-Peer Computing (P2P2004), IEEE Computer Soc. Press.
5. Minsky, M. (1988). *Society of Mind*. Simon & Schuster.
6. *Journal of Artificial Societies and Social Simulation (JASSS)*, available at: (<http://jasss.soc.surrey.ac.uk/JASSS.html>).
7. AgentLink Newsletter 14, research overview article by Steve Phelps and Peter McBurney, available at: (<http://www.agentlink.org/newsletter>).

Forthcoming Event
Socially Inspired Computing
 – engineering with social metaphors
 An AISB Symposium
<http://cfpm.org/sic>