Can Tags Build Working Systems? From MABS to ESOA

David Hales
Centre for Policy Modelling
Manchester Metropolitan University, Aytoun Building, Manchester M1 3GH, UK
+44 161 247 6074
dave@davidhales.com

Bruce Edmonds
Centre for Policy Modelling
Manchester Metropolitan University, Aytoun Building, Manchester M1 3GH, UK
+44 161 247 6479
bruce@cfpm.org

ABSTRACT
We outline new results coming from multi-agent based social simulation (MABS) that harness "tag-based" mechanisms in the production of self-organized behavior in simulation models. These tag mechanisms have, hitherto, been given sociological or biological interpretations. Here we attempt to move towards the application of these mechanisms to issues in self-organizing software engineering – i.e. how to program software agents that can self-organize to solve real problems. We also speculate how tags might inform the design of an unblockable P2P adaptive protocol layer.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents, Multi-agent Systems.

General Terms
Algorithms, Performance, Design, Economics, Experimentation, Theory.

Keywords
Evolution of Groups, Tags, Social Cues, Cultural Markers, Altruism, Cooperation.

1. INTRODUCTION
It would seem that one of the major goals that inspired the agent-based approach (at least initially) was the idea of systems that could self-organize functionality to solve problems [12,20]. Agents would work together, dynamically employing each other's skills to achieve their individual and collective goals [13]. The key here was that a human engineer would not be needed to pre-specify the exact nature of this kind of future interaction (i.e. it would be self-organized).

Although a lot of effort has been directed at the infrastructural base required to realize this goal (languages, architectures, protocols, standards and platforms) much less effort seems to have been directed at the more fundamental issues – i.e. how to make it all "hang-together" [12,13]. More specifically, if we view an agent based system as a kind of artificial society composed of various types of artificial agents then what kinds of mechanisms do we need to implement to allow those agents to form productive teams or institutions? How do we get agents to self-organize their society, rather than requiring the MAS engineer to program everything directly (limiting the complexly possible with current techniques)? Put another way, can we generate and apply a kind of "artificial social science" to the engineering of MAS? What might this kind of "science" look like and how would it be applied?

Over the last ten years a broad multi-disciplinary approach has formed which aims (at least in part) to address this very issue [3]. Often termed "Artificial Societies" [4,2] when applied to possible or abstracted systems and Agent Based Social Simulation (ABSS) when applied to human systems, a growing body of knowledge is being generated. For an overview of current work see contributions to the Journal of Artificial Societies and Social Simulation (JASSS) . Social simulation has always been involved with "messy" systems [24] (i.e. human systems) and consequently has explored mechanisms that may explain order within them.

Within the broad area of Artificial Societies, Multi-Agent Based Simulation (MABS) and Agent Based Social Simulation (ABSS) a number of socially inspired models of self-organized behavior have been explored. In this paper we outline some recently presented "Tag Based Models" (TBMs) [5,6,7,8,18] and indicate how, we believe, the mechanisms identified in these models may begin to be applied to the construction of self-organizing software.

1.1 Tag Based Models
The use of "tags" for the facilitation of self-organization in artificial systems was first explicitly discussed by Holland [11]. Tags are identifiable markings or cues attached to agents that can be observed by other agents. Tags can also evolve or change in the same way that behaviors can evolve and change (for adaptive agents). In a system in which agents evolve artificial genotypes that determine their behavioral traits, tags can be viewed as phenotypically visible portions of the genotype. Early TBM's demonstrated the powerful cooperative effect that tags may foster within some simple repeated interaction scenarios [17].

More recently, TBMs have been advanced that show cooperation and altruism evolving in one-time interactions [5, 6, 18, 22] and facilitating modest forms of agent specialisation and group formation [7, 8]. Also some TBMs demonstrate "reverse scaling" properties where the introduction of more agents produces efficient solutions more quickly [8].

1.2 Paper Outline

Here we outline initial ideas towards the application of the mechanisms identified in these models to the engineering of working information systems that have the ability to self-organize. In another paper [8] we described a simulation model in which we applied a TBM to a simulated robot warehouse-unloading scenario first described by [15, 16]. There we demonstrate that robots controlled by a tag-based mechanism outperformed so called "socially rational" [14] and "selfish" behavior. Firstly we will outline the original scenario and results obtained and then we will described how these mechanisms might be applied in a dynamic peer-to-peer environment to self-organize load-balancing strategies.

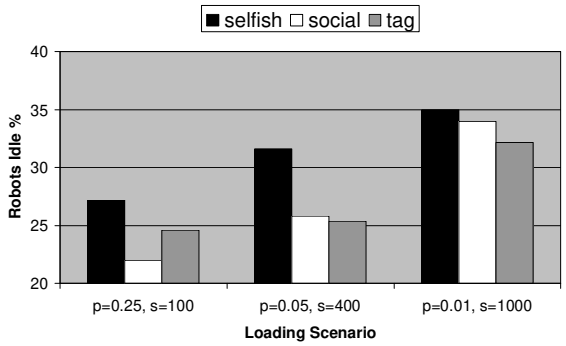
2. THE WAREHOUSE SCENARIO

Robots must work together to service the arrival of deliveries to a warehouse. There are ten loading bays into which trucks can arrive to be unloaded at any time (if the bay is empty). To each bay five robots are assigned to unload the trucks. While working, the robots unload at a constant rate. Robots are rewarded according to how much of the goods in their bay have been unloaded. Robots may help in the unloading of other bays but receive no reward for this. All the bays start empty. If a bay is empty, a truck of size s may arrive with a probability of p in each cycle. The truck stays there until it is empty, then it leaves. Thus the probability, p, is inversely related to the time the bay is empty and the truck size, s, related to the contiguous unloading time.

Each robot has an integer tag T [1..500] which is visible to all the other agents. The only function of the tag is that it is visible to others – it has no other direct significance. The tag-based mechanism is an evolutionary process that acts upon the tags (as well as other behavioral traits) of the robots as they do the unloading. In each cycle each agent can do a fixed number of units (5) of unloading. For each unit, if the agent has a lorry in its home bay it asks another agent for help. First it sees if there is another robot with an identical tag and asks them, if there is not it asks a randomly selected robot. Whether the asked robot responds with help depends upon its strategy, which is composed of two Boolean values: whether to help if it already has its own lorry to unload (L); and whether to help if it does not have a lorry to unload (N) – i.e. if it is idle. Thus the asked robot consults one of L or N depending on whether it has a lorry in its bay and acts accordingly.

At the beginning of each cycle the T, N and L value triples are reproduced into the robot population probabilistically in proportion to the amount that the robot who had them had unloaded in its own bay. With a certain probability of mutation (0.1) these values are replaced with a random new value. Thus successful tags and response strategies are preferentially replicated to the robots for the new cycle. It is therefore likely that if a robot is relatively successful it will be replicated more than once and there will then be at least one other robot with the same tag to cooperate with. No contracts or joint / group utilities are used – the cooperation that emerges is entirely due to the implicit emergence of groups of robots with the same tags. It is a kind of solution to a commons tragedy without central planning [9].

The above outlined tag-based agent decision making is compared to the two 'hard-wired' reference cases, where robots are all "selfish" or "social". In the selfish case robots never help another robot and, hence, only unload trucks in their own bay. In the social case all robots unload from their own bay if there is a lorry there and always help another if there is not and it is requested to do so (i.e. robots will help others if idle and requested to do so). Each simulation was run for 500 cycles (which allows for each robot to unload a total of 2500 units).



Statistics were collected from the runs including the percentage of time the robots were idle.

Figure 1. Total robot idle time for each loading scenario as a percentage of total robot time.

Figure 1 shows for each kind of decision function (or strategy) the total amount of time units wasted by robots sitting idle. An efficient system will minimize this value. Results are given for three different loading scenarios (different values of p and s). When p is small but s is large then deliveries are more sporadic, with lots of help required in bursts, but when p is high and s is low then jobs are arriving at a smother constant rate.

The results given in figure 1 demonstrate a very interesting property of the tag strategy. As expected the selfish strategy performs poorly since idle robots wont help others in other bays, the social strategy performs more efficiently than the tag strategy when p is high and s is low. However, for the two other loading scenarios the tag strategy outperforms the hardwired social strategy. It would seem that the system has self-organized to respond to the sporadic loading scenarios – we are currently investigating the exact dynamics that lead to this very encouraging result. We speculate however, that the tag strategy allows agents to abandon unloading their own trucks in order to help with a newly arriving truck – which would potentially reduce idleness. This is something that the hardwired social agents cannot do. Here we believe we have found an example of the tag system producing a better solution than an intuitively hardwired socially rational solution. Indeed the original purpose of applying tags to the model was to explore an alternative to conventional modes of agent rationality [19, 21].

3. SELF-ORGANIZED P2P USING TAGS

We now outline how the simulation above, which involved robots in a warehouse, might be applied to a peer-to-peer environment in which client/servers need to distribute jobs.

Assume we have a networked environment in which a number of peer client/servers (which we will call agents) handle job requests passed to them from users. The jobs might be information request requiring the searching and filtering of various data sources or even processing jobs requiring the application of computational resources. In any case, each agent can only handle a certain number of jobs at a given time. An agent may, if it is already servicing several jobs, ask another agent to deal with a request. However, the process of locating another agent and making the request would entail computational costs. Let us impose an additional constraint in this hypothetical scenario – assume that credit (however determined) is apportioned from the user to the original agent requested (if the job is performed adequately) no matter which agent actually tackled the job. This latter point simplifies the situation somewhat; since we only require users to connect to a single agent and issue a job request, if the request is met then credit is apportioned in some manner (this could be as simple as some kind of direct payment, or less direct such as causing increasing use of the agent).

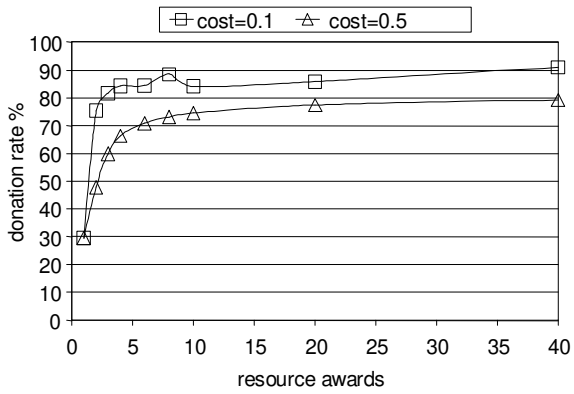
We argue here that if we recast our robots from the warehouse as agents in our network and arriving lorries as job requests from users then we have the basis for a self-organizing load balancing system. How would "reproduction" of a new generation based on credit be achieved in a network peer-to-peer environment? The only condition for this would be that agents periodically compared their credit against that being attained by another and copy the strategies and tags of the other (if the other was attaining higher credit). Such a mechanism would be enough to drive an evolutionarily process. We argue that the simulation results from the warehouse scenario are applicable here – although the minor differences (i.e. each agent receiving direct credit and a slightly different evolutionary mechanism) ultimately require testing in a new simulation model.

Such a tag-based method of job distribution in a dynamic environment would incur minimal overheads – since no central planning, or large amount of information passing is required.

3.1 Specialization & Cooperation

Here we describe a further TBM [7, 8] and in a following section again suggest how the model may be applied to a self-organizing software application scenario. In this model agents are placed in an environment where the formation of groups with internal specialization would produce a beneficial collective effect. Agents are selected at random and awarded resources. However, to "harvest" the resources agents require the correct skill. If they do not have the correct skill they may pass the resource to another agent with that skill (at a cost to themselves) or simply discard the resource. Each agent stores a single integer number representing its skill {1,2,3,4,5} (an agent can only store one skill). Each agent stores two real values: a tag from [0..1] and a tolerance (also in [0..1]). An agent considers another to be in its "group" if the absolute difference in their tag values is less-than-or-equal to the tolerance value of the agent. Agents with higher utility (derived from harvested resources) are periodically copied by others and mutation is applied with some small probability to skill, tag and tolerance values.

During each cycle agents are selected at random and given some number of resources. Each resource is associated with a skill (randomly selected). If the agent posses the correct skill then it will harvest the resource (and gain one unit of utility) but if the agent does not posses the required skill then it will search the population for an agent that is part of it's group with the required skill and pass (donate) the resource (at a cost to itself). It would seem that agents with low tolerances (and hence small or empty groups) would perform well because they would not incur the costs of donation – that is, they would act selfishly. However, a collectively rational solution would be for groups to form with a diversity of skills, that way each member of the group would benefit by passing appropriate resources among each other. After all agents have had some number of awards (this value was



varied see below) agents are reproduced probabilistically based on their relative utility to form a new population (the population size was fixed at 100 for this model).

Figure 2. Donation rates against number of resources awarded to each agent in each cycle.

Figure 2 shows results from simulation of the model. Results are averages over 5 runs each to 30,000 cycles. The number of resource awards is the number of times each agent is given a resource (in each cycle of the simulation) that it can keep, donate or discard. The donation rate is the proportion of times agents donate the resources they are awarded. It indicates the percentage of attempted donations (passes of resources) that result in success (i.e. find a recipient). Notice that the results show that neither pure individual rationality nor collective rationality dominates completely for all numbers of resource reward. However, remember that for high donation rates to occur agents must form groups that contain a diversity of skills. Consequently we are asking a lot more from a simple learning process. Given this we still note that even when the cost of donation is half that of the benefit (cost = 0.5) to the receiving agent, there is still a donation rate of 30% for the lowest number of resources awarded per agent per cycle (this being 1 award). This is far from what individual rationality (individual optimization) would produce (0%) but still means that 70% of resources are being wasted by the system – far from full collective rationality. However when the number of resources awarded to each agent is increased this value quickly increases. When a resource award is set to 5 the donation rate is above 70% for both cost levels. This result indicates that a sufficient number of resources need to be

awarded to agents before they adapt their behaviors. Considering the simplicity of the learning system and the requirement that agents have to produce groups that are internally specialized this is a very promising result.

3.2 Emergent Specialization in P2P

Consider a P2P environment similar to that described previously. In that scenario each agent (client/server node) could process any job that was passed to it by a user. In this sense each agent was a generalist – it could potentially accept any job. The agents could ask for help from others if they were overloaded with jobs. Consider a more realistic kind of scenario in which each agent can deal with some subset of possible jobs, for example some agents may be able to locate information on the web, others may be able to store and retrieve information at a later date others may offer certain kinds of processing (for example, converting a document from one format to another).

We argue the results from the previous simulation indicate that tags may be used to address this problem. A user can pass any agent any job; the agent will then either accept the job (if it has the correct abilities) or pass the job (if it does not have the correct skills) or discard the job. The above simulation results indicate that even if the act of locating another agent with the required skills is costly to the referring agent, this may still evolve, from a process of copying those who are doing relatively better. Again a relatively simple process involving tags (i.e. no central control, contracts or sophisticated redistribution of credit) can produce efficient levels of self-organized structure.

Obviously such a model is very simplistic, more sophisticated agents could learn when to apply their skills to a job (learning of a context or domain of application) [1] and also adapt their behavior to improve their job performance.

3.3 An Unblockable Adaptive P2P Protocol?

We speculate that the cooperation producing process outlined above (for detail see [5]) in PD might be interpreted as the dynamics over time of a set of P2P file sharing systems. Each unique tag represents a unique protocol – i.e. tags are protocols. Agents sharing protocols can interact – cooperation involves productive sharing of computational and data resources (file sharing) and defection involves some non-cooperative activity (not sharing, hogging bandwidth, spyware deployment). Mutation of the tag represents an adaptation or innovation of a protocol. Mutation of a strategy represents innovation of action or even external blocking mechanisms imposed by network providers.

Assuming that agents could intelligently adapt protocols (implemented on-top of the IP and, say, producing permutations of HTTP user / client interaction masquerades such that packet content contained no simple identifiable pattern over time) and communicate those new protocols to their in-group (i.e. those sharing the current protocol) then the tag results obtained in the simulations encourage us to conjecture that this could be a basis for an unblockable adaptive P2P protocol since protocol innovation would be endogenous and fast[1] (assuming thousands or millions of peers). At present this is merely a conjecture and requires further development[2].

[1] Currently systems such as e-donkey, kazza, morpheus and audiognome are exogenously developed (programmed by human programmers) which takes a long time.

4. Conclusion

We do not claim to have delivered a blue-print for self-organizing software applications in this brief paper, rather we wish to demonstrate the at least some ideas coming from the ABSS community have some possible future application in self-organizing information systems. We appreciate that we are far from this goal and that, indeed, we require parallel developments in infrastructural technologies. We only offer an initial step in what could be a productive direction.

However, we argue that via rigorous empirical analysis of ABS and the movement of algorithms from sociological or biological interpretations to information technology solutions and the construction of a series of simulation models – moving closer to implementable systems – some open issues in harnessing self-organization may be productively addressed.

In the first model we showed there that there is no need for complex credit allocations or contracts etc, that tags were sufficient to self-organize collective behavior that (in some task environments) would outperform what would seem to be the intuitively optimal (collective rational) solution (i.e. help when idle, ask when busy). In the second model we demonstrated that a simple form of internal specialization could be attainted using tag processes.

We are encouraged by our initial forays, which seem to suggest that algorithms and techniques embedded in models created for one purpose may be applied for different purposes. Our ambitious hope is that future work may continue to identify, reproduce and test various emergence-generating algorithms producing a kind of typology of emergence techniques and their scope of application.

5. Acknowledgements

The warehouse model was programmed in SDML [23]. Other models were programmed in JAVA2 (www.sun.com). For conversations on the topic of unblockable adaptive P2P protocols thanks go to Jeremy Goslin (www.jeremygoslin.com). We also wish to thank Scott Moss for creating the CPM (http://cfpm.org) and his commitment to the simulation approach.

6. References

[1] Edmonds, B. (2002) Learning and Exploiting Context in Agents. Proceedings of the 1s International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Bologna, Italy, ACM Press, 1231-1238. July 2002

[2] Epstein, J. and Axtell, R. *Growing Artificial Societies: Social Science from the Bottom Up*. Cambridge MA: The MIT Press, 1996.

[2] However, we also conjecture that if such a system could be made to work then it would effectively end attempts to censor the free flow of information through file and resource sharing systems by the powers of darkness.

[3] Gilbert, N. and Doran J., (eds.) *Simulating Societies: the Computer Simulation of Social Phenomena*. London: UCL Press, 1994.

[4] Gilbert, N. and Conte, R. (eds.) *Artificial Societies: the Computer Simulation of Social Life*, London: UCL Press, 1995.

[5] Hales, D. Cooperation without Space or Memory – Tags, Groups and the Prisoner's Dilemma. *Multi-Agent-Based Simulation* (eds. Moss, S., Davidsson, P.), Lecture Notes in Artificial Intelligence 1979. Berlin: Springer-Verlag, 2000.

[6] Hales, D. *Tag Based Co-operation in Artificial Societies.* Ph.D. Thesis, Department of Computer Science, University of Essex, UK, 2001.

[7] Hales, D. Evolving Specialisation, Altruism and Group-Level Optimisation Using Tags. *Multi-Agent-Based Simulation II* (eds. Sichman J., Davidsson, P.), Lecture Notes in Artificial Intelligence 2581. Berlin: Springer-Verlag, 2003.

[8] Hales, D. & Edmonds, B. Evolving Social Rationality for MAS using "Tags". To be presented at the *AAMAS 2003 Conference Melbourne.* To be published by the ACM Press.

[9] Hardin, G. The tragedy of the commons. *Science*, 162:1243-1248, 1968.

[10] Hogg, L. M., and Jennings, N. R. Socially Rational Agents. *Proc. AAAI Fall symposium on Socially Intelligent Agents*, Boston, Mass., November 8-10, 61-63, 1997.

[11] Holland, J. The Effect of Labels (Tags) on Social Interactions. *SFI Working Paper 93-10-064.* Santa Fe Institute, Santa Fe, NM. 1993.

[12] Jennings, N. R. Agent-based Computing: Promise and Perils. *Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden. (Computers and Thought award invited paper) 1429-1436. 1999.

[13] Jennings, N. R. On Agent-Based Software Engineering. *Artificial Intelligence*, 117 (2) 277-296, 2000.

[14] Jennings, N., and Campos, J. Towards a Social Level Characterization of Socially Responsible Agents. *IEE Proceedings on Software Engineering*. 144(1):11-25, 1997.

[15] Kalenka, S. *Modelling Social Interaction Attitudes in Multi-Agent Systems.* Ph.D Thesis. Department of Electronics and Computer Science, Southampton University, 2001.

[16] Kalenka, S., and Jennings, N.R. Socially Responsible Decision Making by Autonomous Agents. *Cognition, Agency and Rationality* (eds. Korta, K., Sosa, E., Arrazola, X.) Kluwer 135-149, 1999.

[17] Riolo, R. The Effects of Tag-Mediated Selection of Partners in Evolving Populations Playing the Iterated Prisoner's Dilemma. *SFI Working Paper 73-02-016.* Santa Fe Institute, Santa Fe, NM. 1997.

[18] Riolo, R., Cohen, M. D. and Axelrod, R. Cooperation without Reciprocity. *Nature* 414, 441-443, 2001.

[19] Russell, S. Rationality and Intelligence. *Artificial Intelligence*, 94(1):55-57, 1997.

[20] Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.

[21] Russell, S. and Wefald, E. *Do The Right Thing: Studies in Rationality*. MIT Press, 1991.

[22] Sigmund & Nowak. Tides of tolerance. Nature 414, 403-405, 2001.

[23] Moss, S., Gaylard, H., Wallis, S. and Edmonds, B. SDML: A Multi-Agent Language for Organizational Modelling. *Computational and Mathematical Organization Theory*, 4, 43-69. 1998

[24] Moss, S. Policy analysis from first principles. *Proceedings of the U.S. National Academies of Science*, Vol. 99, Supp. 3, pp. 7267-7274, 2002.