Malicious behaviors of agents in multi-agent systems: 
minimizing their impact and detection
Research paper
Blazej Bulka
May 19, 2004

Abstract

The paper tries to increase the security and efficiency of multi-agent systems by providing a means to detect malicious agents that either do not contribute to the society or directly do harm. Although multi-agent societies are designed to withstand an external attack, they are much more susceptible to internal attacks, especially if attackers know the structure of the agents’ society very well or if they are important members of this society. The paper proposes a way to prevent attacks exploiting such knowledge of structure by introducing perpetual, local changes to the structure. The changes will make the exact prediction of the global structure of agents very difficult (the agents do not know what changes occurred beyond their vicinity), and therefore they should prevent attacks taking the structure into account. The local span of the changes, on the other hand, is to ensure that they will not interfere substantially with the capabilities of agents to perform their tasks. The introduction of dynamically changing social structure may also impair the efficiency of commonly used trust networks, which are often used in such societies to detect dishonest agents. This paper, additionally, presents a modification to trust management schemes so that they can be used in dynamic societies.

1 Introduction

Reliability is one of the reasons why multi-agent systems (or distributed systems in general) are used. Such systems have decentralized control and resources, therefore there is no single point of failure that would make the whole system unusable, if this point is attacked or suffers from technical problems. However, as of today most of the approaches assume that the attack on the system will be an external one. Less research is directed at protection of the system against internal attacks.
1.1 Dishonest agents

One possible type of an internal attack is when an agent lies about its characteristics, regardless whether these are its advertised skills, goals, or state of some distributed algorithm the agent has reached. A possible aggravated kind of such malicious behavior would exist, if an agent tried not to obey some protocol existing in the agent’s society (e.g., rules used to maintain consistency of variable instantiations in Distributed Constraint Satisfaction Problems). I expect that usually, if the system is able to produce any valid solution at all, the quality of solution may suffer or its performance may significantly degrade. There may be also scenarios and approaches, where a single malicious agent may not be able to cause significant harm (at least on a short-term basis).

An interesting question is: What is the “critical mass” of agents doing wrongful actions that would cause a society of agents to collapse? One can imagine a situation when an agent advertises a skill that it does not possess. It does not necessarily mean that the society of agents to which this agent belongs will suffer in performance because of this untrue advertisement. There may be other agents that have the required skills or the lying agent may be able to find a subcontractor (either within the group or outside of the group). In cases where agents are evaluated on performance of their groups, the strategy of not telling the truth all the time may even be beneficial.

1.2 Agent failures

The impact of an agent failure on the society of agents depends on many factors like the structure of the society, the approach it uses, and the redundancy in skills and data. (There is ongoing research in the Maple lab about the significance of the structure of the society on the reliability of the whole system.) This problem is related also to the problem of dishonest agents described in the previous subsection because a malicious agent may penetrate the society of agents and coordinate an internal attack with an external one. This could be achieved by either analyzing the structure of the society and directing an external attack at weak spots or by becoming an important part of the society and then simulating a failure. Ironically, flexibility of self-organization in agent societies could be used to leverage such an attack. Therefore it would be useful to investigate some organization strategies in agent societies (e.g., how agents decide which organizational structure to choose based on current situation; what may be a stimulus to initiate a change of structure, and how agents gain places within the structure).

1.3 Practical implications

Recent years have brought emergence of heavily distributed systems whose nodes may be perceived as autonomous agents. Examples of such systems may be peer-to-peer systems (e.g., applications like Gnutella, Freenet, or JXTA library) or attempts to run ad-hoc networks, in which low power devices act as proxies for each other, depending on their physical location. In all of these systems one can see a trend toward elimination of all centralized elements that
cannot be replaced without a disruption of the whole system. The centralized components that have been the most difficult to convert into distributed versions include naming services, information routing within a network, and authentication. Most of the mentioned systems tried to provide alternatives of these services, which varied widely in the quality.

The initial success of the Gnutella file sharing application proved that it is possible to create a working, fully decentralized solution that will be resistant to different kinds of external attacks at its structure. Its performance was even more amazing given the simplicity of the underlying communication protocol. The deployment of the application revealed also the problems I described in the previous subsections. First of all, the application was not immune to network fragmentation. Therefore a failure of a highly-connected node often led to loss of communication and even worse, there was no direct possibility for the users to detect this condition and subsequently to recover from it. Second, some nodes abused the purpose of the network by either spreading detrimental content or erroneous advertisements about the content. Since the network did not allow to identify the wrongdoers or did not provide any service to establish positive or negative reputation, the society was effectively helpless against these practices.

The problems presented are also important to the field of web-services, which are envisioned to enable dynamic service discovery in order to combine them automatically into more complex services.

1.4 Outline of the proposed solution

The solution I propose to solve tries to deal with two problems: preventing of an internal attack at the structure of the society and detecting dishonest agents in the society. The first problem, the exploitation of the social structure of agents, may be prevented by turning a static structure into a dynamic one, where social relations periodically have to change. (Social relations determine with whom an agent may cooperate or with whom it may communicate.) The changes are intended to confuse malicious agents because they are not able to track all the changes in the society, only the local ones. Whereas this surely prevents these agents from attacking important parts of the structure, or from becoming such parts, care must be taken that honest agents can still perform their tasks. (Obviously a structure that is changing chaotically cannot provide any benefits to either malevolent agents or to the benevolent ones – such a solution would be like protecting a computer against attacks by disconnecting it from the network.)

An answer to the second problem may be establishing a trust network among agents, in which no central authorities exist. The proposed trust network solution would be a modification of existing trust network solutions in which the trust level is not only based on witnesses who are evidence of trustworthiness, but also on active querying of the neighbors to detect inconsistencies in the environment. The modifications of the existing solutions are necessary because they often fail to address the issue that dishonest agents may lie about their previous interactions, or may try to protect the other malevolent agents within their own group. Additional justification for these modifications is that existing solutions would be affected
adversely by the dynamic changes of the society proposed in the first solution. (The both solutions are meant to be complementary rather than exclusive to each other.)

2 Trust management in multi-agent systems

The notion of trust and its propagation in social structures is currently an active research area, especially in environments where there are no central authorities. An important and still unsolved problem is that it is not known whether trust is a transitive property (i.e., if agent $A$ trusts agent $B$ and agent $B$ trusts agent $C$ should agent $A$ trust agent $C$ and to what extent?)

The paper by Yu and Singh [2003] presents a model of reputation based on positive and negative evidence collected from witnesses. Each agent is supposed to maintain the references to the witnesses, with whom it has interacted in the past, so that the next collaborators will be able to verify its trustworthiness. Agents are not restricted only to following the given references, they can also query the agents in their neighborhood (immediate and indirect as bounded by arbitrary depth limit). The opinions given by agents may be positive, negative, or an agent may not have an opinion at all – either because it has not had any experiences with the agent being verified or it is not certain enough to express an opinion. (The Dempster-Shafer theory of evidence used by the authors states that there is no causal relation between a positive hypothesis and the negative one.) After having queried all relevant agents, the verifying agent is supposed to combine the evidence (possibly inconsistent) by using a weighted sum. The weights are determined by weighted majority algorithm and are adjusted if the outcome turns out to be incorrect. Yu and Singh performed also an interesting analysis on impact of incorrect ratings, both in case when agents unconsciously exaggerate their ratings (positively or negatively) and in case when they maliciously express unjust opinions.

An interesting idea of trust having internal structure is presented in the work by Castelfranchi et al. [2003]. According to them trust has many different properties like for example, competence, accessibility, or danger, which contribute to the overall trust value. Additionally they allow multiple sources to be combined that contain information about an agent. The choice of the most important factors that influence the trust and merging them together is done by use of different heuristics and the use of fuzzy values.

The presented solutions may not necessarily perform well in the scenario I propose to solve. The second solution [Castelfranchi et al., 2003] is not only too domain dependent but it is also not known, how it would perform in an environment, where agents expressing their opinions are allowed to lie. The first solution, although it turned out to be very similar to mine, relies on the social structure being stable for a longer time, in order to learn appropriate weights for the weighted majority algorithm.
3 Proposed solution

This section contains the detailed description of the proposed solution. The first subsection describes the main idea, which prevents exploiting the social structure of agents by malicious members of the society. Unfortunately applying this solution may create problem for some types of services needed by agents, and this could make them mutually incompatible with the proposed solution. An example of such a service is trust management, which is also used to detect malicious agents in the society (a related goal). Making these two techniques exclusive would be a serious drawback of the proposed solution, and therefore a possible modification to trust management is described in the second subsection.

3.1 Preventing exploiting the social structure of agents

The social structure of agents can be exploited by a malicious agent in one of two ways. First, the agent may direct an external attack at a weak point in the structure. The other, more intricate way of exploiting the structure is becoming an important node within a structure and either simulate its own failure, lie to its neighbors, or secretly modify or drop messages from other agents routed through this agent. (This assumes that agents may act as proxies between other agents.) The result of such actions may be either degrading performance of the whole society or some kind of fragmentation of the social network.

The described problem may be prevented if agents periodically and collectively change their local structure of neighbors (e.g., by voting). Such changes would prevent a malicious agent from becoming an important node paralyzing the society for a longer time. Moreover, if a malicious agent should deliberately cause a failure of an important node, the society would recover during the next change. The locality of changes ensures that the operation of the whole society is not much disrupted, since the all the voting agents know the new structure and can forward messages from non-local agents.

The proposed solution assumes existence of local groups of agents (connected by neighbor relation), and such groups may be perceived by non-local agents. The groups will have agents performing administrative functions (e.g., connectivity with other groups). It should be also possible to address and deliver messages using name of the administrative function rather than the name of the particular agent.

The change period mentioned above is necessary to ensure that no group of agents will be able to “freeze” their current roles for a longer period of time. Although the period itself may be negotiable (e.g., to replace an important agent after a failure in a timely manner), it should not be possible for agents to extend this period over some predefined threshold as this could disable the security mechanism.

Impact of changes on the performance The changes are supposed to prevent any agent from tracking the global structure of the society. Whereas all agents know the changes that occurred in their vicinity, they do not have information about other parts of the society. This enforces the rule that all remote interactions between agents should be initiated using the
proposed administrative function based addressing. Additionally agents interacting remotely have no information about the location of their peer in the society. The proposed restrictions should lead to increased security because a single, malicious agent can exploit only the structure of its local community, without causing a significant disturbance to the society as a whole. Overcoming these restrictions would require a whole group of cooperating malicious agents to be distributed evenly in the society. An even distribution, however, is also made difficult by these rules, since the global structure is not known. The malicious agents may also have problems communicating to coordinate their actions if their locations are remote.

The claim that local changes will not disrupt the actions of the honest agents in a significant way depends on the fact that all agents in the local group know the previous structure and the new structure and can forward messages appropriately, and therefore the changes are not significant at the local level. If an agent could send messages to another agent before the change, this ability is preserved after the change. Unless a failure or an attack occurs, the local structure becomes irrelevant to the local society. (In case of attack the reliability of communication still depends on the structure.)

The remote agents, on the other hand, can access their peers using the administrative function based addressing. This assumes that the peer’s identity is not relevant, and agents are exchangeable as long as their skills are equivalent. For most services this assumption is a valid one, but there is a group of services that depends on the identity of the peer and does not allow its substitution. I expect that most services that does not allow substitution are services meant to provide authorization and security to the agents’ society like for example, trust management.

**Voting strategies** Efficiency of the solution will depend to a high degree on the exact voting strategies in the local societies that control the process of local changes. The most effective voting scheme is not yet known and its details will probably be domain dependent. The voting scheme needs to address the following issues in more detail:

- The administrative functions that need to exist in the society. (Possibly also the abilities that need to be possessed by agents to apply for a function.) If agents were allowed freely to propose their own functions this could lead into a security breach because malicious agents could coordinate their actions and exchange information about the structure of the society using the administrative function based addressing scheme. (They could propose their own functions, which only they would apply for.)

- The size of the local group, the policy for accepting new members, and the policy for leaving the group. Otherwise a group of malicious agents could use the group joining and leaving mechanism to disperse them evenly in the society. (Even totally unintelligent agents can be dispersed almost evenly in an environment by an external, chaotic factor; such a process would be similar to diffusion.)
3.2 Tracking honesty of agents and ostracism toward violators

A society of agents may defend itself by decreasing the role and participation of agents, who have been detected to be dishonest, until they are effectively excluded.

The approach presented in this paper is a modified approach of the one presented in the work by Yu and Singh [2003], which would probably not work well in a system, whose social structure changes often. They rely on the Weighted Majority Algorithm (WMA) that adjusts weights for witnesses based on the correctness of their previous testimonies. The perpetual changes of the society structure also change the set of available witnesses as the time passes and therefore it would impair the learning of the WMA algorithm. The proposed contributions to this method, which are described below, are keeping a list of positive witnesses by each agent, who can confirm its credibility, and periodical querying of the environment to detect possible inconsistencies.

Establishing trust network  In this approach each agent will maintain a trust table $T$ toward the agents, with whom it is either currently in cooperation or the cooperation has been recently terminated. The agent would also remember all the agents who can verify its own trustworthiness (table of positive references $R$) based on previous successful interactions. Meeting of two agents that have no information about each other would involve assigning neutral values of trust in their respective trust. The initial value of trust could be modified then in the following way:

- For each positive reference that can be verified (i.e., the agent who issued the reference can be contacted and is directly trusted, as indicated by $T$) the trust level is increased.

- The agent being verified undergoes a “background check” among the current neighbors of the verifying agent by asking them for opinion, if the newcomer is known to them at all. A neighbor’s opinion would be also weighted by the level of trust of the verifying agent to this neighbor. The limited trust to neighbor’s opinion has been introduced to prevent malicious agents from organizing into more influential groups.

Agents would be also required to store some responses received from their current neighbors for some period of time. The actual period of time would be determined by the level of trust toward a particular neighbor – the lower the level of trust, the more data would have to be stored. Periodically (exact value also dependent on trust level) an agent would ask its neighbors for responses received from the agent being verified. These responses would be then analyzed to detect inconsistencies that may be presumptive evidence of dishonesty. The importance of such evidence would be additionally weighted by the level of trust toward the neighbor. Such evidence (or lack of thereof) would be a reason to decrease the trust level of the agent being verified (or increase respectively). An additional, useful source of data to detect inconsistencies in the neighborhood would be traffic analysis of the messages being proxied by an agent. (The information about interactions occurring among other agents can be collected without understanding the semantics of the sent message and this information may be used to infer partial knowledge about the mutual trust between interacting parties.)
Potential problems with this solution  The solution presented above uses first-hand knowledge only in determining the trust level. An alternative solution including varying level depending on the size of the society is also possible. This solution will be probably refined after performing a more extensive literature survey in the area of trust networks. Moreover, detecting inconsistency may not be possible in all domains, and therefore there is a risk that this approach may have limited applicability.

4  Methodology

This section describes the test environment I plan to build (along with test scenarios and possible simplifying assumptions) to verify my claims described above. Later I describe the base line approach and the evaluation criteria that will allow to compare the efficiency of my solution relatively to the efficiency exhibited by the base line solution.

4.1  Test environment

Properties of the environment  To test the methods described in the previous section I plan to create a simulation environment. The environment will enable agents to organize themselves into local social structures and to participate in them by voting. An important property of the test environment will be the fact that agents may only communicate with their direct neighbors. More remote communication would only be possible through other, intermediate agents acting as proxies. All agents in the test environment should communicate asynchronously to avoid blocking of multiple agents by one unresponsive agent (it can be unresponsive deliberately). The asynchronous delivery of messages and their processing by the agents will also prevent “spurious agent-agent correlation” ([Axtell, 2000]; a programming artifact emerging due to sequential activity of agents), what may introduce error into results.

The assumption that agents will be able to organize themselves into social structures relies on the fact that they know some agents, which in turn know other agents. This may, however, not always be the case, especially at the beginning of the simulation (no agent knows any other agent) or during fragmentation of the social network. Therefore there is need to provide some kind of “dating service” that will allow agents to get to know each other. The service would select randomly another agent and give its address to the requesting agent. Any more complex “search” would not be allowed. (Agents are supposed to socialize themselves mostly by using the structure they managed to create, not through any other external means.)

Local structures  Agents are expected to form small societies, which in turn will aggregate into higher level structures. The exact shape of these societies, voting strategies is yet to be investigated. (These are the details I plan to work on during making research on this problem.) Each local society will have agents having some administrative roles. Although the exact list of roles that will exist in the local structures is not yet known (in most flexible design vision agents should decide themselves, which functions they need), but some already seem to be necessary:
• Gathering tasks from outside (explained in the next paragraph).
• Connectivity with other groups (possibly a “neighboring groups”).
• Management of agents joining and leaving the group.

Other administrative functions that do not seem necessary but their may be useful include trust management function or voting leader function.

(Please note that a single agent can form a group just by itself and it can have all administrative functions at once.)

Problem solved by agents The evaluation of the proposed solution will be based on a scenario of task allocation in a multi-agent systems. Randomly selected agents having appropriate administrative role in their society will receive “external offers” (i.e., introduced from outside of the society) to execute some task. The structure of the tasks will be given a priori and it may consist of subtasks, constraints between subtasks, and a set of skills required to execute a task. It is expected that the agents receiving these offers will not be able to accomplish these tasks by themselves and therefore they will be forced to hire other agents who have the required skills (skills should be redundant in the society). The agents being hired, in turn, will be allowed to lie about their skills in order to get credit for belonging to a successful group. (Lying is supposed to be a beneficial strategy because they can hope that they will not be elected to do the task, they advertised they can do, or that they will be able to find an appropriate subcontractor.) After the successful execution of the task the hiring agent will receive payment from its external employer and will pay the subcontractors or the whole local society. (Unless the hiring agent is a dishonest one.)

Note: the design of the testbed described above has been inspired by the simulation environment created by John Simmons and Matt Gaston (related to [Gaston and desJardins, 2003]).

4.2 Evaluation method

I plan to evaluate, whether the proposed methods work at all by comparing them first to the most naive behavior, this is taking no precautions in an environment with dishonest agents, gradually increasing their number till the society effectively collapses (“critical mass of dishonest agents”). Such analysis would allow not only estimation, whether the methods work at all (performance of the system is better and the “critical mass” to create a collapse has to be greater) but also an estimation of the overhead of the method. It is likely that for very small number of agents in the society the cost of execution of additional mechanisms may be prohibitively large.

Measuring efficiency and overhead One possible metric for measuring efficiency of the proposed solution may be counting the total number of tasks that agents are able to execute in a unit of time. Another rational measure of efficiency would be counting the percentage
of tasks completed as opposed to the number of tasks that were accepted by the agents. The both methods should reflect the degradation of performance as the number of dishonest agents increases. (It is expected that malicious agents will interfere with a group’s communication abilities or will make the group unable to execute the task, for which it has applied.) The first method, however, has a drawback because it relies on a global notion of time (time units). Because of the asynchronous communication among agents, this approach may not always be suitable (no guarantee of response time and dependence). The other method may be seen as more suitable but it is not entirely clear, how to define an uncompleted task if we allow tasks without deadline dates. (A task that has not been executed in any given period of time can be perceived either as a failure of the team to execute it, or that task’s execution has been deferred until a later time.) An interesting alternative to measure efficiency, would be to calculate how much profit agents can collect by being dishonest. (This measures not only the incentive to be dishonest, but also how much profit has been lost by the honest agents.)

As it was stated earlier, part of my evaluation method will involve measuring the overhead caused by the precautions related to dishonest agents. The overhead may be measured by counting the amount of all messages exchanged globally by the agents compared to the amount of messages related to safety precautions, voting and structure changes.

**Definition of collapse** The evaluation method described above involves measuring the moment of collapse of the agent society. I expect that decreasing effectiveness of the society will not be a linear process, but it will be possible to represent it as a curve that is either convex or concave (roughly). Such curve would have exactly one inflexion point that may be perceived as the beginning of the collapse. Intuitively, for convex curves (as perceived from the origin of the coordinate system) the inflection point means that the initial rapid degradation has begun to slow down, and for concave curves it means that the initial small impact of dishonesty has just rapidly increased. The convex shape of the described curve would mean that an agent society is very vulnerable to any dishonesty and the concave shape would mean that it is capable of withstanding substantial number of malicious agents within it. Therefore the degree of convexity or concavity may be another measure of evaluation of the quality of the proposed method. (Especially if the application of the method would convert a normally convex curve into a concave one.)

### 5 Related work

The problem I try to solve is related mostly to coalition formation (methods of formation of societies by the agents), trust management, and the impact of the topology of the agents' society on the overall efficiency of the society. This section describes the most relevant parts of these areas.
5.1 Coalition formation

Coalition formation approaches in general try to group agents together, mostly to optimize the overall efficiency of the groups, satisfy individual goal of the agents, and minimize potential conflicts within the group (maintaining group stability). The main difference between the solution presented in this paper and most of the coalition formation techniques is that the latter usually neither emphasize impacts of dishonesty on the performance, nor they support distributed computation, where no global knowledge is accessible. The beneficial aspect of coalition formation is that it cares about improving efficiency of the group, and therefore some of the ideas from this area may be incorporated into the voting schemes of the presented solution.

5.1.1 Kernel based approaches to coalition formation

Kernel based approaches usually use a numerical score that is maximized by appropriate assignment of agents to groups. Such score may be either an individual or joint utility function, or satisfaction of individual goals of the agents. (In case of satisfying needs of individual agents within the society usually Pareto-optimal solutions are desirable.) Kernel based approaches tend to look for equilibrium of the power distribution within the group. Unfortunately finding an exact, optimal solution is computationally exponential and therefore approximation approaches have to be used (e.g., Polynomial Kernel-Oriented Coalition Algorithm (KCA) [Klusch and Shehory, 1996]).

Many kernel based approaches are inappropriate in real-world applications, because they have unrealistic assumptions like for example, the exact knowledge of all parameters a priori, or assumption of honesty of the participants. The work by Blankenburg et al. [2003] describes a more realistic model, in which all values are considered to be inexact (fuzzy). The model has been obtained by modifying KCA algorithm to accept fuzzy values. Some important simplifications, however, were made with respect to the archetype, like lack of support for distributed computation, and all agents share the same beliefs about the degree of inaccuracy of the data.

Paper by Kraus et al. [2003] presents another way of removing the necessity of possessing global knowledge and knowing exact values of all parameters. They propose the use of estimated values and heuristic to iteratively optimize the values. An important property of this solution is that it is so called any-time solution (i.e., computation may be interrupted at any given moment and a partial result can be retrieved). This property may be important, if applied to the problems where structure needs to change from time to time (e.g., due to agents leaving or joining the system).

5.1.2 Trust based coalition formation

An alternative approach to kernel based coalition formation has been presented in work by Griffiths and Luck [2003]. Instead on focusing on equal distribution of score, payoff, or power this approach takes into account the current motivations of agents and the current trust level
toward prospective coalition members. Aggregating agents with similar goals (motivations)
and trusting each other is supposed to create teams that will last longer than the teams
created on the base of maximizing utility function. The expected longer life-time of a team
created in such a way can be explained that utility function’s value is usually computed only
for the current task, and therefore long term cooperation is rarely taken into account. On
the other hand common goals of agents mean that they potentially can cooperate for longer
time (until the goals should change). The other factor, the trust, is important because it
determines agent’s risk of being involved in the cooperation that can be in reality different
than previously advertised.

5.2 Interaction topologies in multi-agent systems

Investigation of the interaction topologies in multi-agent systems has been an active re-
search area in the recent years. The topologies of such systems can vary from very reg-
ular ones like lattice (very few neighbors for each node, long paths between nodes, but
easy analysis of the structure) to totally random graphs (each node has high degree, short
paths between nodes because many shortcuts exist, but the whole structure is difficult to
analyze). A popular intermediate structures between these two extremes are small world
graphs and scale-free graphs [Gaston and desJardins, 2003; Walsh, 1998; Delgado, 2002;
Axtell, 2000], which are supposed to model well the social structures existing in the hu-
man society and existing complex technical infrastructure like the Internet or electrical power
grid. Small world graphs can be constructed by creating any regular graph (e.g., a lattice)
and then introducing random disruptions to the structure (“rewiring”) to create shortcuts
between nodes, which have been proven to dramatically reduce the diameter of the graph
(and successively the average path length). The scale-free graphs are created by probabilistic
growing of the graph, where new nodes are attached with greater probability to the nodes
with a higher degree.

The paper by Walsh [1998] is one of the papers that presents the impact of small world
topologies on the efficiency of search. A more economic approach to evaluation testing of
different topologies is presented in a paper by Axtell [2000]. An interesting part of this paper
deals also with random activity of agents within the network (“agent activation regime”) and
random changes of the structure. The important conclusion for my approach is that small
changes in the society can be achieved quite easily by totally random agents’ choices, but such
technique is definitely inappropriate for larger societies and changes. (This gives justification
for more intelligent voting schemes to change the structure.)

The last paper I would like to mention is the paper by Gaston and desJardins [2003], in
which the efficiency of various topologies is compared using task assignment scenario. The
conclusions are that scale-free graphs perform distinctively better than the random graphs,
which in turn are still better than small world and lattice graphs.
6 Future work

The current approach does not concentrate at all on the way how the agents vote, and what strategies they use to elect their agents that will perform administrative functions. Additionally, if there are multiple strategies available, how do agents choose the one that should be used? Should the choice of the strategy depend on the current structure of the society or on the estimated number of malicious agents?

The reputation management scheme presented here is currently simplified. One can think of more efficient schemes of establishing trust rather than leaving each of the agents on its own and forcing it to track the reputation of others by itself. It would be interesting to investigate, whether it is not beneficial to have special administrative functions of agents that attempt to track the trust toward greater number of agents. Agents having this administrative duty would periodically change, according to the general schema of function that has been introduced to prevent abuse. Another extension to the proposed reputation schema, could be to adopt some of the notion of structured trust described in work by Castelfranchi et al. [2003] for example, to distinguish between intentionally wrongful agents and the agents that became overloaded with tasks and cannot complete them all. Such overloaded agents may appear, if the dynamically changing structure should create nodes in the social structure that have extraordinarily high degree.

The proposed approach may not also scale well as the society of agents becomes larger. If this fact is confirmed, a possible extension of the proposed method could be to agglomerate smaller groups of agents into bigger groups. The relation between the super-groups and the sub-groups could be then established through a system of representatives. However, it is not clear how regular and hierarchical such structure of the society should be. A recent research [Gaston and desJardins, 2003] shows that societies having structures similar to regular graphs and totally random graphs may have problems in effective solving their tasks, whereas so called scale-free graphs created in a more structured but still random way may yield much better performance. Additionally if agents agree on transition to another social structure to support expansion or shrinking of the society, what is the vulnerability of the society during the transition process?

Finally, prevention and detection of social network fragmentation still remains unsolved in the current solution.

7 Conclusions

This paper presented proposed solutions to problem of detection of agents (or elements of a distributed system in general) that do not contribute to the society or outright do harm to some members of the society, and prevention against such agents. The presented problem needs to be solved, if we want to achieve a fully decentralized, distributed systems that are able to adapt themselves to changing conditions (e.g., in systems that are supposed to act autonomously in open and heterogeneous networks like Internet).

The presented solutions focused mostly on agents that plan to attack from inside, especially
if they plan to take advantage of knowing the society’s structure or to become an influential member of the society. The key idea is to introduce some dynamism into the structure of the society so that analysis or longer-term plans about becoming an influential member are either infeasible or carry too much uncertainty. (Alternatively this approach may be viewed as introducing some noise that obscures analysis of the structure.) The changes to the structure are planned to be introduced in such way that they will not disrupt the function of the society in general. Additionally a specialized way of establishing trust and keeping track of malevolent agents in the fully distributed society without implicit authorities and with everlasting changes of agents’ neighborhood was also presented.

References


