Building and Using Social Structures: A Case Study Using the Agent ART Testbed

ELISABETTA ERRIQUEZ, WIEBE VAN DER HOEK, and MICHAEL WOOLDRIDGE,
University of Liverpool

This article investigates the conjecture that agents who make decisions in scenarios where trust is important can benefit from the use of a *social structure*, representing the social relationships that exist between agents. We propose techniques that can be used by agents to initially build and then progressively update such a structure in the light of experience. We describe an implementation of our techniques in the domain of the Agent ART testbed: we take two existing agents for this domain ("Simplet" and "Connected") and compare their performance with versions that use our social structure ("SocialSimplet" and "SocialConnected"). We show that SocialSimplet and SocialConnected outperform their counterparts with respect to the quality of the interactions, the number of rounds won in a competition, and the total utility gained.

Categories and Subject Descriptors: I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms: Design, Experimentation, Performance

Additional Key Words and Phrases: Agents, trading competition, trust

ACM Reference Format:
Erriquez, E., van der Hoek, W., and Wooldridge, M. 2013. Building and using social structures: A case study using the agent art testbed. ACM Trans. Intell. Syst. Technol. 4, 2, Article 25 (March 2013), 20 pages.
DOI = 10.1145/2438653.2438660 http://doi.acm.org/10.1145/2438653.2438660

1. INTRODUCTION

Trust and reputation are key issues in the multiagent systems domain. As in human societies, software agents must interact with other agents in settings where there is the possibility that they can be exploited. This suggests the need for computational models of trust and reputation that can be used by software agents, and accordingly, much research has investigated this issue over the past decade [Abdul-Rahman and Hailes 2000; Rubiera et al. 2001; Castelfranchi and Falcone 1998; Jurca and Faltings 2002; Marsh 1994; Schillo et al. 2000; Teacy et al. 2006; Huynh et al. 2006; Sabater and Sierra 2002].

One important question is what sources agents can use to build their trust of others upon. For example, agent a can base its trust or reputation of agent b using experience of previous interactions between the two; or agent a might ask a third party c about its opinion regarding b. An important additional source of trust is to use information about the social relationship (here called the *social structure*) between agents [Sabater and Sierra 2002]. If a and b are competing for the same resources, for example, this may negatively affect the way they trust each other. Similarly, if agents a and b are

An abstract of this article appeared as Erriquez et al. [2010].

Authors' addresses: E. Erriquez (corresponding author), W. van der Hoek, M. Wooldridge, Department of Computer Science, University of Liverpool, L69 3BX United Kingdom; email: e.erriquez@liverpool.ac.uk.

DOI 10.1145/2438653.2438660 http://doi.acm.org/10.1145/2438653.2438660

likely to have complementary resources, and their cooperation would benefit both, it seems likely that they would be more inclined to trust each other.

Although models of social structure have begun to be considered in models of trust and reputation [Sabater and Sierra 2002], to date, implementing social structures, and hence properly evaluating their added value and validating them, has not been done. And, most importantly, the issue of how a social structure evolves does not appear to have been considered in the literature. These are the issues we address in this article.

In order to compare different models for trust in agent communities, and to provide an experimental standard, the Agent ART testbed [Fullam et al. 2005] was developed. This testbed models an art appraisal scenario, where agents can offer their expertise to rate a painting, or give their opinion about the expertise and trustworthiness of other agents. The Agent ART testbed is a platform for researchers in trust to benchmark their specific trust models. We took two of the agents from the Agent ART testbed, "Simplet" [Krupa et al. 2009] and "Connected", and enhanced them with a representation of a social structure, and algorithms to build and refine this social structure. We named the two resulting agents SocialSimplet and SocialConnected, respectively. Then, we ran a number of competitions in which, among others, both Simplet and SocialSimplet and Connected and SocialConnected participated, and we evaluated and compared their performance. In particular, we measured the number of successful interactions for each of them, the utility gained, and the number of rounds they won in a competition. On all of those measures, the *social* agents performed better than their asocial counterparts, which encourages us to explore our social structures further and to extend our experiments to other scenarios.

In summary, in this article we: (*i*) describe how agents can gradually build and update a representation of the social relationships and structure of their society, (*ii*) provide an implementation of such representation of the social structure, and (*iii*) test and evaluate the result of the use of such a structure in a trust model. The remainder of this work is organized as follows. Section 2.1 gives a preliminary presentation of the social structure and a survey of research in this area. Section 2.2 gives a brief overview of other models that use social information and Section 3 describes the scenario used for the evaluation. Section 4 explains the process used by the agent to build and update the representation of the social structure. Section 5 presents the evaluation and discusses the results we obtained. Finally, Section 6 presents some conclusions and suggestions for future work.

2. RELATED WORK

2.1. Social Structure

Social network analysis is the study of social relationships between individuals in a society, defined as a set of methods specifically developed to allow research and analysis of the relational sides of these structures [Galaskiewicz and Wasserman 1994]. These social networks are represented using graphs called *sociograms*. A different sociogram is normally built for each type of social relationship examined, for example, *kinship* or *friendship*. According to the relationship considered, the graph can be directed or undirected, with or without weighted edges. However, for the purpose of this article, we will use a simplification of these multiple sociograms. We will use a single sociogram that we call a *social structure*. Using multiple sociograms allows the agents to be linked to others in more than one social relationship at the same time. Conversely, using a single sociogram allows the agents to be linked together in just one social relationship at any given time. In Section 4.2, we will explain more in detail how we create the social structure and how we take into account the different types of social relationships.

In the multiagent systems context, social structure analysis can play a vital role. For example, the search for relevant information involves finding the right sources: the agents who have the required information or expertise. Thus, social network analysis can be an important tool in discovering these relevant information sources. The interactions between individuals in the society produce the addition of a new link in the social structure. Agents are aware of the presence of other agents in the society because of their direct interactions, using services or asking opinions. However, interactions and recommendations are also useful in predicting the relationships among agents. An agent can infer from an indirect recommendation that a witness agent has used the target agent (i.e., the agent the recommendation is about) as a service provider. This is because it is assumed that if the witness agent has an opinion about the target agent, it is because they have interacted in the past.

It might be considered debatable, but in practice, we surely use this kind of "social logic" every day. Asking for a recommendation about a mechanic or restaurant employs exactly this type of reasoning, taking into account the trust we have in the person we are asking and his trust of the mechanic or restaurant.

Similarly, these trust relationships are used by computational agents to minimize the uncertainty the agents have while interacting. Although direct interactions are perhaps the most reliable source of information to compute trust values, one cannot assume that each agent, at any stage, has interacted with every agent that is relevant. Therefore, an agent might not be able to form an opinion, based just on direct experiences, about every agent in the society. That is why "reputation" information from third-party agents, called witnesses, can complement information gathered from direct interaction.

Although the aim of a recommendation from a third agent is to evaluate the trustworthiness of the target, the agent asking the recommendation is also able to draw a link, in its social structure, between the witness agent and the target agent, thus building a more complete view of its environment. This simple intuition is used in our approach to build the agent's social structure.

The utility of large amounts and different sources of information is also considered in Quattrociocchi et al. [2008]. In fact they show, through experiments, that using different sources of information contributes to uncertainty reduction in situations where trust is required. This is exactly what we aim to show in this article, but we concentrate on the specific contribution given by information from social relationships. Our tests aim to show that a social structure as source of information, together with the common sources, can help agents in making more informed decisions in uncertain situations.

The ultimate goal of our work is to allow agents to make better choices of the partners to interact with. The goal of partner selection in social networks has been studied in social sciences for a long time. For instance, the embeddedness of the interactions in the social structure [Granovetter 1985] has been identified as a main ingredient in explaining the evolution of cooperation [Macy and Skvoretz 1998].

Also Takács et al. [2008], in their work, assert that the social ties created by social networks influence member behavior through perceived social gains and rewards for themselves. In particular, they allow their agents to strategically revise their relations. Similarly, our work tries to give more information for the agents to base their strategic decisions.

Initially, theoretical models, predominantly based on game theory, only considered one-to-one games but in subsequent research, these models have been extended to allow for situations in which there is a network of agents who can communicate about the behavior of a particular agent [Weesie et al. 1998; **?**; Buskens and Raub 2002]. The latter emphasizes the connection between learning (basing trust on past experiences) and control (possibilities of sanctioning an untrustworthy agent). In Weesie et al. [1998], it is argued how the density of this network of agents has a positive effect both on trust and on how quickly information can reach other agents. This sociological work on trust and cooperation is relevant to our setting, although our work is focused on artificial computational agents in which the trust relationships are given in a very specific way, and we test our model in a given computational testbed. Flache and Hegselmann [1999] base their modeling on cellular automata, where the decisions of the automata depend on their neighbors (so the network model distance between agents).

2.2. Social Models

In the last twenty years, many computational trust and reputation models have been developed [Abdul-Rahman and Hailes 2000; Rubiera et al. 2001; Castelfranchi and Falcone 1998; Jurca and Faltings 2002; Marsh 1994; Schillo et al. 2000; Teacy et al. 2006; Huynh et al. 2006; Sabater and Sierra 2002]. Some of the models are derived from Marsh's formalism [Marsh 1994], where trust is used to assess the probability that an agent will keep its promises.

An example of a simple and intuitive trust model is Liar from Muller and Vercouter [2004]. Liar has a reputation model built from an agent's own interactions, from its observations, and from recommendations sent by other agents. It uses a violation detection process working with incomplete information. Internal rules are set to help in the detection of violation, such as rules finding contradictory information. One of the agents used in our test uses Liar as inspiration for its internal trust model.

However, in this section we concentrate in giving a brief overview of some trust and reputation systems that use social information in their computational model. Some of these models integrate the concept of agent neighborhood in their models.

An interesting decentralized reputation systems is the one based on the referral system described by Yu and Singh [2002], who developed a mechanism to identify information sources, called witnesses, based on each agent's knowledge and past behavior, through use of their contacts. Each agent in the system maintains a list of acquaintances (other agents that it knows) and their expertise. When the agent needs some information, it sends a query to a certain number of agents in its list who will try to answer the query, if possible. If they can't, they will send referrals pointing to other agents that they believe may have the information, basing this belief on those agent's expertise.

Other trust-based network models include Histos [Zacharia et al. 1999]. The Histos model can deal with both direct and indirect information, and the reputation value is a personal subjective value assigned by each individual. The ratings are represented as a directed graph in which nodes represent agents and edges carry information about the most recent reputation rating given by one agent to another.

Our work has some similarities to these two models. Similar to our approach, Yu and Singh [2002] model tries to create a network of friends from which to extract more information about the society, while Histos [Zacharia et al. 1999] represents the society in a graph similar to our social structure. However, while in Histos edges represent trust relationships, in our model edges represent interactions.

The models presented so far try to incorporate a social dimension in the way they compute the trust value, either using a network of trust relations for weighting the other agents' opinions or using information about the role the agents have in the society. Although agents are assumed to never lie, they can hide information or bias it to favor their goals.

In Regret [Sabater and Sierra 2002], it is claimed that the model goes one step further and considers that agents can also lie. It presents a slightly different use of the social network from the previous models. Regret has a model of trust and reputation based on different sources of information; in Regret's case, these sources are: individual, social, and ontological. The social dimension includes information about the experiences of other members of the evaluator's group, or neighborhood, which is assumed to be a group of agents with some shared knowledge. However, Regret takes advantage of social relations between agents to overcome the lack of information that the agents may face. It is assumed that each agent owns a set of sociograms that describe the relationships linking the agents in the society. In particular, unlike the other models, the edges in the sociograms do not carry information about past interactions or trust among agents, instead they represent the relation that the linked agents might have in a market-like scenario, like for example, collaboration or competition; further details will be given in Section 4. The agents use the sociograms to find witnesses, to decide which witnesses will be consulted, and how to weigh those witnesses' opinions. However, what was not addressed in Regret, and what we think is important, is that such sociograms in the end need to be acquired somehow. This is one of the aims of our article.

Moreover, Regret does not show how witnesses can be located. Fire [Huynh et al. 2006], to overcome this problem, employs a referral process in which agents help each other to find witnesses based on their expertise. In Huynh et al. [2006], it is also argued that a robust trust model should integrate a number of information sources in order to produce a comprehensive assessment of an agent's likely behavior. Therefore it uses a modular approach that integrates up to four types of information sources, according to availability: interaction trust; role-based trust; witness reputation; and certified reputation. The notion of neighborhood is used in Fire's witness reputation module for searching for relevant witnesses. This module is based on Yu and Singh's [2002] referral system, enabling agents to share referrals for the location of relevant information. However, Fire, unlike Regret, assumes that the other agent will always be honest when sharing information. Although we don't assume that agents will always be honest, we don't directly deal with the problem. As will be better clarified in Section 4.3, it will be the trust model of the agent to deal with potentially false information, according to its own strategy.

Therefore, to summarize, in this work, we propose techniques that can be used by agents to initially build and then progressively update a structure representing the social relationships that exist between agents in the light of experience. Although we use a particular scenario to validate our hypothesis, the way the agent uses to build and update its social structure can be easily generalized to other "market-like" scenarios.

In the next section we give a brief description of the scenario used to validate our approach.

3. TESTBED AND RUNNING EXAMPLE

The scenario for our running example is the Agent ART (Agent Reputation and Trust) testbed [Fullam et al. 2005]. The Agent ART testbed game is intended to support the comparison of different strategies of agents as they act in combination. We decided to use the Agent ART testbed as an experimental platform because it covers relevant trust research problems and helps researchers find a solution via unified experimentation methods, so it is a versatile experimentation tool. Moreover it provides objective metrics through which it is possible to compare and validate different trust models.

The context of the testbed is the art appraisal domain, in which agents function as painting appraisers with varying levels of expertise in different artistic eras. Each appraiser agent competes against every other agent in the system. Clients request appraisals for paintings from different eras; if an appraiser agent does not have the expertise to complete the appraisal, it can request, at a price, opinions from other appraiser agents. Appraiser agents may also purchase from each other reputation information about other appraisers. Appraiser agents must decide when, and from whom, to request opinions and reputation information to generate accurate appraisals for clients. Appraisers receive more clients, and thus more profit, for producing more accurate appraisals. The winning appraiser agent is selected as the appraiser with the highest bank account balance.

In the Agent ART testbed, asking for reputation information has a cost. Using the social structure to select only agents that are more likely to provide meaningful information could translate to a cost saving. Once the information is gathered, the opinion from the witness agent selected is weighed taking into account the social relationships linking the agent with the witnesses and the target agent.

4. BUILDING THE SOCIAL STRUCTURE

4.1. Basic Concepts from Graph Theory

The social structure is represented as an undirected graph. Therefore, in this section we present some notions of graph theory that we use in our model (for a more complete and detailed background on graph theory, see Wallis [2007]).

Let $G = (V, E)$ be a graph where V is a set of vertices of G and $E \subseteq V \times V$ the set of edges of G.

G is a directed graph iff E is a set of directed edges (otherwise it is called undirected graph). Note that it is equivalent to consider that an undirected graph is defined by a symmetric relation E on $V \times V$.

A connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and to which no more vertices or edges (from the larger graph) can be added while preserving its connectivity. (This is known as a maximal connected subgraph [Wallis 2007].)

Sometimes, a connected graph G can be so sparse that the removal of a single vertex will disconnect it. Such vertices are quite important in our work.

A vertex x is called a *cutpoint* in G if $G \setminus \{x\}$ contains more components than G does; in particular if G is connected, then a cutpoint is a vertex x such that $G \setminus \{x\}$ is disconnected [Wallis 2007]. Figure 1 shows an example.

From a sociological point of view, a cutpoint can be seen as indicating some kind of Local Centrality (LC). In our context it can be considered as the agent centralizing the largest amount of information among the agents in the components it connects. A similar concept is expressed from the central points. There are various measures of the centrality of a vertex within a graph that determine the relative importance of a vertex within the graph. We use the *degree centrality*, defined as the number of edges incident upon a vertex. Figure 2 shows an example. Degree is often interpreted in terms of the immediate risk of a node for catching whatever is flowing through the network, such as a virus, or in our case, some information.

4.2. Identifying the Social Relationships

The concept of using a social structure to enhance a trust model was initially proposed in a computational agent in the `Regret` model [Sabater and Sierra 2002]. In Regret, an agent owns a set of sociograms that show the social relations in the society. These sociograms are not necessarily complete or accurate. However, `Regret` does not propose a way for the agent to build sociograms.

The main contribution of the present article is to show how the social structure can be built by each agent in the society and show, via evaluation, how using the social structure in a trust model can improve the performance of the trust model. Using the Agent ART testbed, we are able to use agents from the past competition and enhance their internal trust models with the social structure. As the agents interact with other agents, they gather information about interactions and relationships in order to build the network of agents and to better understand their social environment.

As in `Regret` [Sabater and Sierra 2002], we consider three main types of social relationship.

—*Competition (COMP)*. This is the type of relation found between two agents that pursue the same goals and need the same (usually scarce) resources. This is the kind of relation that could appear between two sellers that sell the same product or between two buyers that need the same product.
—*Cooperation (COOP)*. This relation implies the exchange of sincere information between the agents and some kind of predisposition to help each other if possible. In other words, we assume that two agents cannot have at the same time a competitive and a cooperative relation.
—*Trade (TRD)*. This type of relation is compatible either with cooperative or competitive relation. It reflects the existence of some commercial transactions between two agents, without a distinctive competitive or collaborative behavior.

To identify the type of relationship between two agents in our Agent ART testbed, we use the concept of expertise. The Agent ART testbed assigns to agents different levels of expertise for each art era. The agents, during the game, can ask each other information about their levels of expertise. The agents are providers of a service which is the evaluation of a painting. However, they are, at the same time, consumers of this service, when their level of expertise is not sufficient to evaluate a particular painting. This means that agents with similar expertise compete in the society for the same market share. Agents with different expertise are more likely to cooperate because they will need each other's services.

The Agent ART testbed scenario is designed so that agents are globally in competition for client share, but able to cooperate on single appraisals. They are providers of a service, appraising a painting of a particular era, however, they might need to buy that same service from other agents who are more expert than them in that era.

Formally, the social structure is defined as a undirected weighted graph $G = (V, E)$ where:

—V is a finite set of vertices that represent the agents in the society,
—$E \subseteq V \times V$, is the set of the edges that correspond to links between agents, and
—each $(a, b) \in E$ has associated with it a weight w_{ab}.

The weights associated on each edge will be used to identify the social relationship between the two agents linked by that particular edge. To define this weight, we introduce the concept of *expertise distance*. To compute how similar or different the levels of expertise between two agents are, we use Euclidean distance. Each era is considered as a dimension in an n-dimensional Euclidean space, where n is the number of eras. The total number of eras is defined in the games setting of the Agent ART testbed. The values of the different levels of expertise identify points in the Euclidean space. We can, therefore, define an agent x as a vector $v = a_1, a_2, \ldots, a_n$ where $a_i \in 1, \ldots, n$ is the level of expertise for era i, $i \in 1, \ldots, n$ and n, which is the total number of different eras, is set in the game configuration settings.

The Euclidean distance $d(a, b)$ between two n-dimensional vectors, or agents, is given by

$$d(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}.$$

This distance is called the *expertise distance* and it is the weight assigned to the edge linking the agents a and b.

In the case where the agent does not have information about the expertise of the other agent in a particular era, it is assumed the agents have the same expertise. If the agent assumes that the unknown value is at the low end of the range of expertise's

ALGORITHM 1: An algorithm for computing the Social Structure *Soc* for agent x.

```
begin
    $Soc \leftarrow \emptyset$;
    while game is not over do
        // if $x$ asks $y$ for a painting appraisal
        if $askedAppraisal(x, y)$ then
            // an edge between $x$ and $y$ is added if not already present
            if $(x, y) \notin Soc$ then
                $Soc \leftarrow Soc + (x, y)$;
            // the weight of the edge is updated
            $weight[(x, y)] \leftarrow computeDistance(x, y)$;
        // if $x$ asks $y$ for information about $z$
        if $askedRepInfo(x, y, z)$ then
            // an edge between $y$ and $z$ is added if not already present
            if $(y, z) \notin Soc$ then
                $Soc \leftarrow Soc + (y, z)$;
            // the weight of the edge is updated
            $weight[(y, z)] \leftarrow computeDistance(y, z)$;
    return $Soc$
```

value, this might artificially increase the distance measure. Setting it to the same value will be like not considering that dimension in the distance measure.

Agents who are a small distance apart have similar expertise and are considered to be in a social relation of *competition*. The further apart the agents are in the space, the more they depend and cooperate with each other, finding themselves in a social *cooperation* relationship. When agents are not a clearly small or large distance apart, we consider them to be in a *trade* relationship. This relationship does not indicate an unclear situation where the agent does not know to do, but it simply means that the two agents, according to their expertise, could occasionally cooperate or compete. In the evaluation, we use a singleton value, as parameter of the experiments, to assess the relationship, called *expertise threshold*.

4.3. Building and Mantaining the Social Structure

The steps used by the agent to build the social structure are as follow. Initially, the agent x starts an empty social structure. Every time x directly interacts with another agent y, asking for a painting appraisal, an edge is created between x and y. Every time the agent x receives reputation information from a witness agent y about a target agent z, an edge is created between y and z. This is because, as mentioned before, it is assumed that an agent can infer from an indirect recommendation, or reputation information, that the witness agent has used the target as a service provider. The edges linking the agent that owns the social structure and other agents are a result of direct interactions. The edges linking the other agents in the social structure are the result of indirect interaction, or recommendation.

Algorithm 1 describes in pseudocode the procedure used to describe building the social structure.

As explained earlier, the use of the social structure in the trust model is inspired by `Regret` [Sabater and Sierra 2002]. However, the aim of our experiments is to analyze the impact of the social structure on a general trust model. Therefore the trust model of the agents selected is not changed to conform to the `Regret` trust model. Hence, it will be the trust model of the agent to deal with potentially false information, according

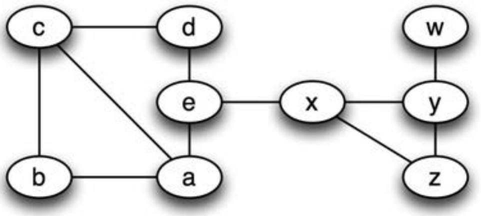

Fig. 1. Cutpoints: e, x and y.

ALGORITHM 2: An algorithm for assigning weight on the edge between agents x and y.

```
begin
    foreach $era \in Eras$ do
    // Gather information about expertise of agent $x$ from the Agent ART testbed
       Simulation Engine
     $expertise_x[era] \leftarrow expertise(era, x)$;
    // Gather information about expertise of agent $y$ from the Agent ART testbed
       Simulation Engine
     $expertise_y[era] \leftarrow expertise(era, y)$;
    // if we don't know both values, we assume them to be zero
    / if $expertise_x[era]$ and $expertise_y[era]$ is null then
        $expertise_x[era] \leftarrow 0$;
        $expertise_y[era] \leftarrow 0$;
    else
    // if we don't know one of the values for that era, it is assumed the agents
       have the same expertise
     if $expertise_x[era]$ is null then
        $expertise_x[era] \leftarrow expertise_y[era]$;
    else if $expertise_y[era]$ is null then
    $expertise_y[era] \leftarrow expertise_x[era]$;
      // Compute the Euclidean Distance between the two vectors of expertises
     $w \leftarrow computeEuclideanDistance(expertise_x, expertise_y)$;
    return $w$;
```

to its own strategy. The agent trust model is integrated with the use of the social structure as follows. We use the social structure to find witnesses to ask for reputation information about other agents, to decide which witnesses will be consulted, and how to evaluate those witnesses' opinions.

The first step to calculate a witness reputation is to identify the set of witnesses (W). The initial set of potential witnesses is the set of all agents that have interacted with the target agent in the past. Starting from these agents, the social structure is filtered, leaving the portion that contains the selected agents. The result of this first step is a subgraph of the initial social structure. This graph can be formed by more than one connected component.

In a large society, the set of witnesses can be very big, and agents with frequent interactions are likely to have a considerable amount of shared information that tends to unify their way of thinking [Pearl 1988].

Algorithm 3 shows, in pseudocode, the steps required to compute the set of witnesses for a particular target t.

The agents in the final set of witnesses, W, will be asked for reputation information about the target agent.

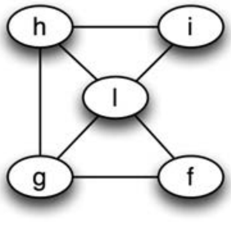

Fig. 2. Central point: l.

ALGORITHM 3: An algorithm for computing the set of witnesses W, from the Social Structure Soc, for target agent t.

```
begin
    $Ws \leftarrow Soc$;
    // Remove from $W$ edges whose vertices are not linked to $t$
    $remove - all(W,\ x,\ y)\ where\ x \neq t\mathbf{and}y \neq t$;
    // Identify the connected components of $W$
    $Comp(Ws) \leftarrow connectedComponents(Ws)$;
    // For each component, finds cut-points or central points
    foreach $W_i \in Comp(Ws)$ do if $hasCutpoints(W_i)$ then
        $W \leftarrow W + cutPoints(W_i)$
    else $W \leftarrow W + centralPoints(W_i)$ return $W$;
```

According to Figures 1 and 2, the set W would be formed by agents e, x, y, and l. If W or the initial set are empty then the agent will behave in the same way as without the use of the social structure. This situation can happen, for instance, at the beginning of the game, where a sufficient number of interactions has not taken place yet to allow the social structure to have meaningful information. The social relationship linking the target agent and the witness agent affects the way the reputation information is considered. For example, the agent is likely to discard reputation information from witness agents who are in strong competition with the target agent.

At every time-step, the social structure, and hence the weights on the edges, are updated according with new information received from the witness agents. This includes adding new links between the agents or modifying the weights in relation to any expertise change. This helps the social structure to be dynamic and to reflect changes in behavior from the agents in the society.

We have integrated this strategy in two agents that were previously developed for the Agent ART Competition. From an initial examination of the implementation and documentation of the agents, we have noticed that many of the agents participating in the Agent ART Competition didn't implement the reputation module, available in the Agent ART testbed framework, in their trust model. They only relied on direct interactions, because the total population in the competition was not large, allowing them to interact directly with almost every other agent. The only agents implementing the reputation module were Simplet and Connected.

Simplet's trust model is inspired by `Liar` [Muller and Vercouter 2004], whose model has been introduced in Section 2.2.

Connected's trust model creates and updates at every time step a list of "friends". These friends are selected calculating the accuracy of the previous paintings' evaluations. Connected also has internal checks and different thresholds to decide how accurate the evaluation is.

The next section will explain the process used for the evaluation, explaining the parameters and the metrics used. We also provide an analysis of the results.

5. EXPERIMENTS AND ANALYSIS

One of the contributions of this article is to provide a systematic evaluation of the use of the social structure, implemented as explained in the previous section, in a trust model.

Seven different configurations were developed. The parameters of each configuration are shown in Table I. The configurations contain parameters for the simulation environment, the Agent ART testbed, and for the configuration of the social structure in SocialSimplet and SocialConnected. For each configuration, 50 runs were carried out, as shown from the parameter *#games*. All configurations have a constant set of agents consisting of those used by the 2008 Agent ART Competition (excluding Agent Uno [Murillo and Munoz 2008]), plus SocialSimplet and SocialConnected. In this phase of our experiments, although the agents are allowed to lie, we tried to reproduce a fair competition environment. Therefore, excluding Agent Uno was necessary because its trust model uses knowledge of the design of the Agent ART testbed framework to tune the parameters of its trust model. These parameters are available to programmers because they are described in the documentation of the Agent ART testbed, however, information regarding these parameters is not available to the agents during the competition, unless the designer explicitly incorporated it during the coding phase. Therefore, in our opinion, even if the parameters are available to designers, they should be deliberately ignored for the sake of not biasing the trust model.

The configurations can be grouped in three subcategories. The first category includes configurations A, B, and C. The aim of this first group of configurations is to test the performance of the agents in settings with different dynamism. Parameters such as *#era to change* and *amount of expertise change* set, respectively, how many eras the expertise value will be changed for at each time step, and the amount by which it will change. As mentioned before, the era represents the period of time a particular painting belongs to, and the expertise value represents how confident the agent is about its knowledge in a particular era. Therefore, by changing the level of expertise the agent has to react to variation in the other agents' responses which might be due to different expertise rather than strategy. This first group reflects the test setting used in the official Agent ART Competition[1].

The second category includes configurations D, E, F, G, J. For this group we consider the intermediate value of dynamism for the parameters *#era to change* and *amount of expertise change*. The aim of this group of configurations is to test, in particular, the "social" agents with respect to their performance in a setting where the cost of communication among agents is removed. Since the agents using the social structure make heavy use of reputation and expertise[2] requests, it is possible that their final performance is affected by these costs. In this group of configurations, we also vary the total population, going from 1 instance of each agent in the game to 10 instances for each agent. In configuration J, we also increase the number of times teps. With these variations, we want to test if the social structure proves to be more useful in a larger society and over a longer period of time.

The final category is comprised of configurations H, I, L, M, and N. Again, in this last group we consider the intermediate value of dynamism for the parameters *#era to change* and *amount of expertise change*. We also keep the cost of the communication

[1] From the Agent ART Testbed Web site. http://megatron.iiia.csic.es/art-testbed/competition2008.htm

[2] The expertise requests are called in the Agent ART Competition setting *certainty request* because the agent can communicate its level of certainty about an opinion on a particular era.

Table I. Configuration Settings for the Evaluation

Parameters	Configurations												
	A	B	C	D	E	F	G	J	H	I	L	M	N
# games	50	50	50	50	50	50	50	50	50	50	50	50	50
# timesteps	90	90	90	90	90	90	90	200	90	90	90	90	90
# eras	20	20	20	20	20	20	20	20	20	20	20	20	20
avg client/agent	20	20	20	20	20	20	20	20	20	20	20	20	20
client fee	100	100	100	100	100	100	100	100	100	100	100	100	100
opinion cost	10	10	10	10	10	10	10	10	10	10	10	10	10
certainty cost	1	1	1	0	0	0	0	0	0	0	0	0	0
reputation cost	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0	0
# opinion msg	3	3	3	3	3	3	3	3	3	3	3	3	3
# certainty msg	10	10	10	10	10	10	10	10	10	10	10	10	10
deny self opinion	true	true	true	true	true	true	true	true	true	true	true	true	true
# eras to change	1	3	3	3	3	3	3	3	3	3	3	3	3
amount expertise change	0.05	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
# instance of each agent	1	1	1	1	2	5	10	10	1	1	1	1	1

initial reputation discard	true	true	true	true	true	true	true	true	false	true	true	true	true
expertise threshold	15	15	15	15	15	15	15	15	15	2	5	10	20

Note: Top section contains settings for the Agent ART testbed and the bottom section contains the settings concerning the social structure of SocialSimplet. Dark grey cells contain parameters changed for that configuration.

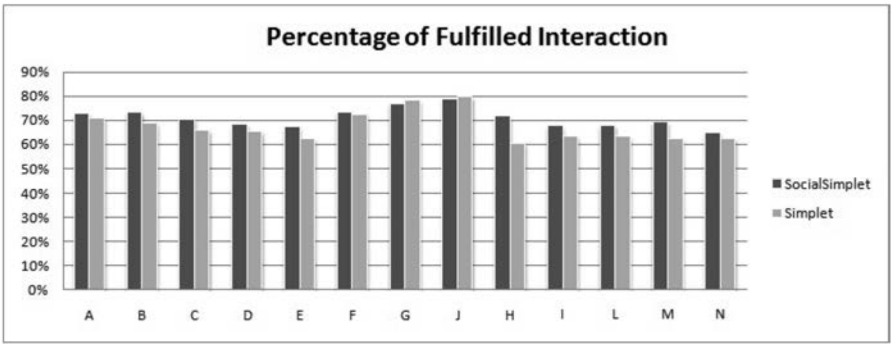

Fig. 3. Percentage of fulfilled interactions for all configurations for Simplet and SocialSimplet.

among agents to zero. In this category, we test the performance of the agent by varying parameters of the social structure (bottom Section of Table I), such as the *expertise threshold* used to identify the social relationship between two agents in the social structure. This set of configurations also helps to fine-tune performance linked to these parameters in the Agent ART testbed.

The dark gray shaded cells in Table I identify the important parameters whose values were changed for the given group of configurations.

Several metrics are used to evaluate performance. Each metric is observed for Simplet and Connected and their counterparts using the social structure, which we call SocialSimplet and SocialConnected. The values are used to evaluate and compare the performance of the agent with its social copy using the social structure. The metrics considered are:

—the total utility, as in the Agent ART Competition,
—the number of games won, grouped for each configuration setting,
—the number of fulfilled and violated interactions, compared with the total number of interactions.

Every *interaction* is intended as an exchange of information between agents. This information can be an opinion about a painting or reputation information about another agent. These two types of information can be considered as the two services provided by the agents in the environment. This final metric is particularly important in assessing the value of the social structure because it reveals how many times the trust model has been successful in selecting a trustworthy agent. To assess whether an interaction has been fulfilled or violated, the internal trust model of each agent is analyzed. In both Simplet and Connected, the agents use the difference between the true value of the painting and the opinion provided by other agents in previous interactions. The true value of each painting is revealed after every interaction, allowing the agent to calculate this value. The difference in the values, together with the expertise of the agent providing the opinion in the considered painting's era, allows the agent to decide if the interaction was fulfilled or violated. If the difference is above a certain threshold, the interaction is considered violated. This threshold may be different in Simplet and Connected because it is set in the internal trust model of each agent. The purpose of these experiments is not to compare the trust model of Simplet versus Connected, but to compare each of these agents with their social counterpart. Thus this threshold is not considered as a parameter of the experiments. Since each agent and its social copy have the same trust model, this threshold will have the same value in both agents, meaning the comparison is not affected by this.

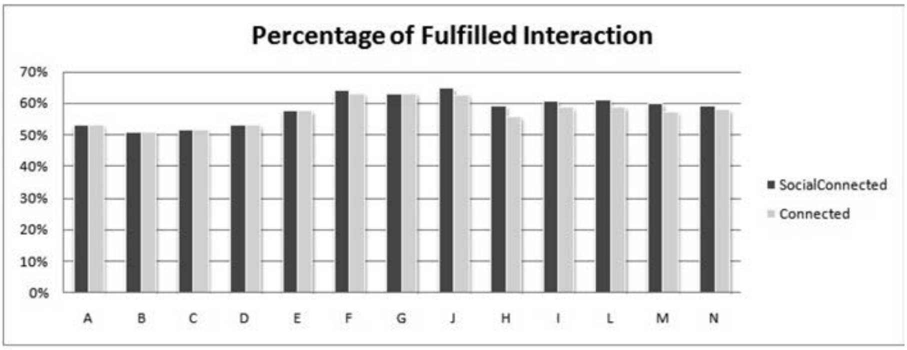

Fig. 4. Percentage of fulfilled interactions for all configurations for Connected and SocialConnected.

In Simplet, in Figure 3 we can observe the percentage of fulfilled interactions averaged over the fifty runs for each configuration. Although our calculations show that, overall, there is an improvement of only 3.64%, the total number of interactions is considerably larger in SocialSimplet, as shown in Figure 5. The results show that SocialSimplet has, on average, 8200 more interactions than Simplet. Thus the effective number of fulfilled interactions is substantially larger. This improvement in the number of fulfilled interactions clearly shows that the internal trust model of the agent benefits from the information derived from the social structure. This suggests that the trust model with the social structure can more accurately assess which agent is more likely to be trustworthy. Since the social structure is updated after every time step, SocialSimplet is only marginally affected by the change of dynamism in the configurations A, B, and C, going from a 72.6% of fulfilled interactions in configuration A, to a 70.5% in configuration C. Simplet seems to suffer slightly more because of this change, losing a little more than 5% over the three sets of games. A similar trend may be observed in Connected. Figure 4 shows an improvement of nearly 2%. Although this value is smaller than the percentage reached in Simplet's case, in this case there is a very large increase in the number of interactions by SocialConnected, with respect to the number of interactions in Connected, as shown in Figure 6, with over 350,000 more interactions, on average, over all configurations. We have analyzed the reason for this behavior. We observed that the trust model of Connected is quite complex, and sometimes the agent ends up not asking for much information, because no agents satisfy its complex constraints. However, in SocialConnected, where the social structure is used for selecting the more trustworthy agents, most of the time, the agent manages to select a set of agents to interact with. This results in more interactions. In conclusion, the difference in the number of fulfilled interactions for SocialConnected is, on average, more than 2 million in each configuration. Also, SocialConnected is only marginally affected by the change of dynamism in the configurations A, B, and C, going from a 53% of fulfilled interactions in configuration A, to 51.66% in configuration C.

Configurations F and G show the highest percentage of fulfilled interactions among the other configurations, in both SocialSimplet and SocialConnected. This means that the social structure is particularly useful in those societies with a large number of agents, where direct interactions are not always possible with every agent in the society. In configuration J, SocialSimplet nearly reaches a total of the 80% of fulfilled interactions against a total of more then 130 thousand interactions, hence having more than 102, 600 successful interactions. SocialConnected reaches 65%, having more than 1 million successful interactions. This shows that the accuracy of the social structure improves over time, because after every interaction the social structure gains more

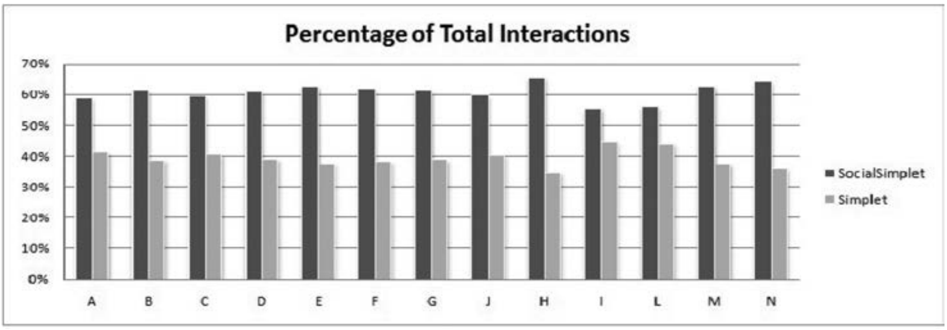

Fig. 5. Percentage of total interactions for all configurations for Simplet and SocialSimplet.

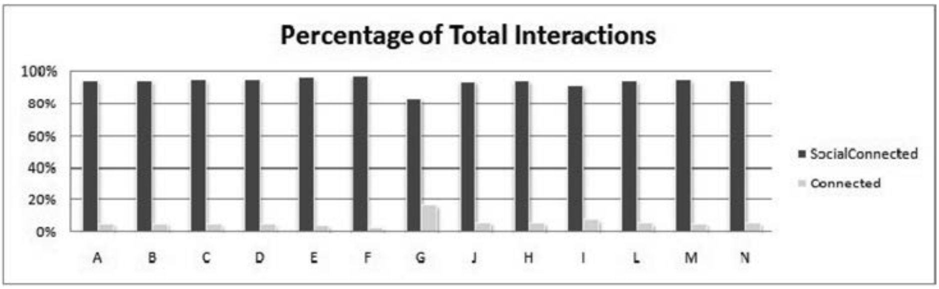

Fig. 6. Percentage of total interactions for all configurations for Connected and SocialConnected.

information about the society. Therefore in very long games, such as the runs in configuration J, the agent can refine the social structure and obtain a better perception of the environment it inhabits.

In configuration H, we consider that, at the beginning of the game, agents cannot yet have meaningful information about the other agents in the society. Therefore we do not add links between agents in the social structure when we receive reputation information at the beginning of the game, if the agent thinks they are initial default values. In Figure 3 and Figure 4, we can observe that ignoring this initial information does not seem to adversely affect the performance of SocialSimplet nor SocialConnected. This is because, since the social structure is updated at every time step, the agent can correct any wrong assumption. From the set of games in configurations I, L, M, and N, we can learn that a good expertise threshold value seems to be one in the range 10.081 to 15.081. The expertise threshold is used to assess the types of social relationships linking the agents. We can also note that in configurations D, E, F, and G, where the reputation and expertise requests have no cost, there is a progressive improvement in the utility gained, in line with the increasing total population, in both SocialSimplet and SocialConnected.

On the other hand, when considering total utility, SocialSimplet performs better over all the different configurations, but only by a small margin. Figure 7 and Figure 9 show, respectively, the percentage of games in which SocialSimplet wins, and the margin by which it wins. This shows that SocialSimplet wins very often (almost 100% of the games), but by a small margin. This is due to the client allocation share strategy, which is affected by the accuracy of the final appraisals produced. The agent chooses a total amount, representing the time taken to examine the painting, to be spent in generating its own opinion about a painting value. This amount affects the accuracy of the generated final appraisal; this parameter is a strategic decision on how the

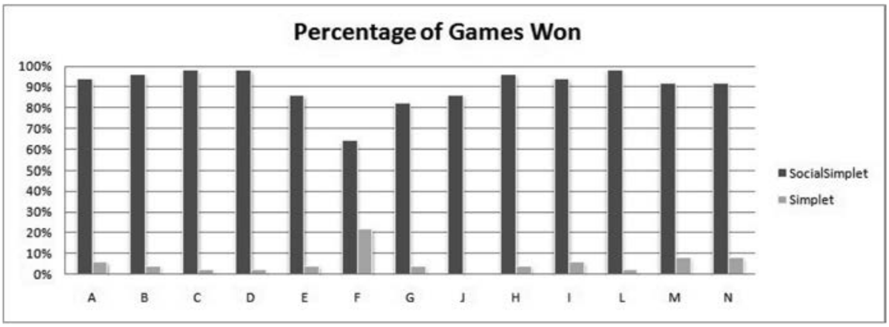

Fig. 7. Percentage of games won for all configurations for Simplet and SocialSimplet.

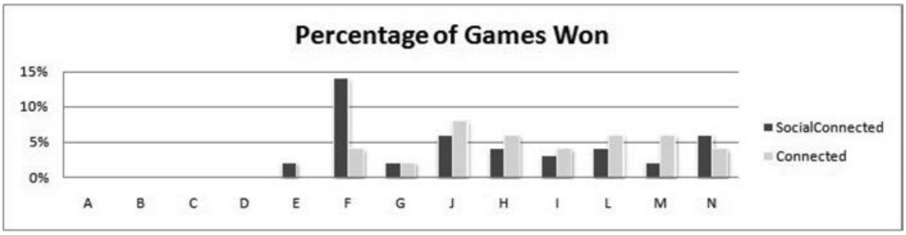

Fig. 8. Percentage of games won for all configurations for Connected and SocialConnected.

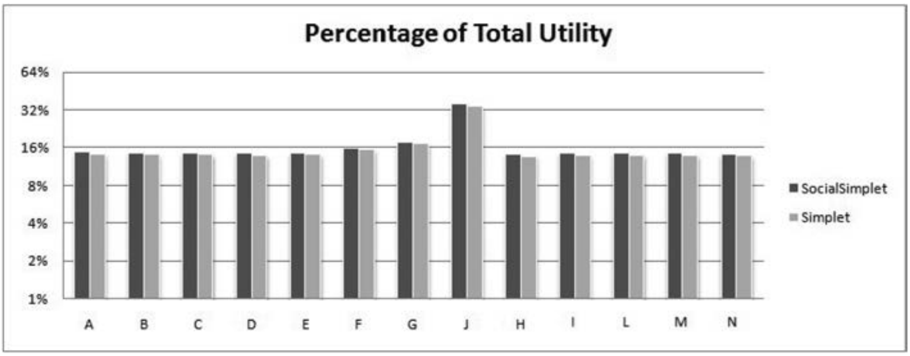

Fig. 9. Percentage of total utility for all configurations for Simplet and SocialSimplet.

agent manages its money, and it is not affected by the changes made in SocialSimplet. In other words, the way Simplet administers its money is not part of the trust model, even if it is affected by it to an extent. More details about this issue are presented in Section 5.1. The margin of improvement in the accuracy of the appraisals produced is due to the improved selections of other agents who are asked for their opinions, but it is harder to appraise given that the total utility is determined largely by the client allocation share strategy, which is the same for Simplet and SocialSimplet in this evaluation. Considering that the only difference in the two models, Simplet and SocialSimplet, is the use of the social structure, it seems reasonable to infer that any improvement in the performance is due to the social structure.

In Connected and SocialConnected we see similar behavior. The difference in the total utility gained, between the two agents, is more evident, both in the case where Connected earns more and vice versa. We can observe in Figure 10 that

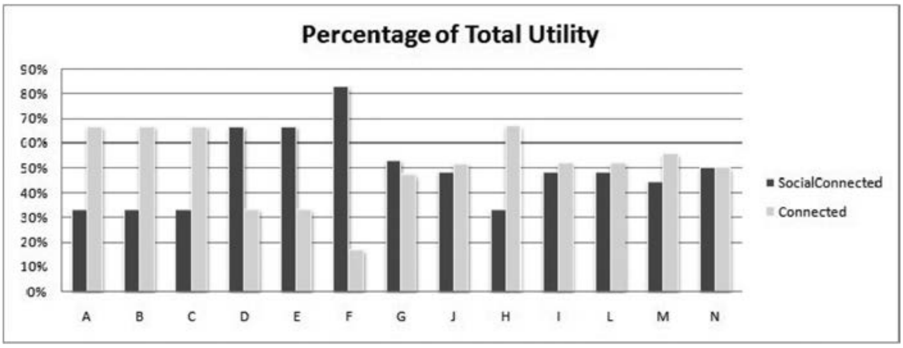

Fig. 10. Percentage of total utility for all configurations for Connected and SocialConnected.

SocialConnected gains much more utility when the costs linked to the formation of the social structure are removed, as shown in configuration D to N. This means that the agent has more money available to invest in the generation of a more accurate appraisal, hence affecting the client allocation share strategy and giving Social Connected the possibility, in the next time step, to improve its performance.

5.1. Remarks on the Agent ART Testbed

The results presented in the previous section demonstrate that the use of a social structure can offer some benefits. However, some remarks concerning the Agent ART testbed are necessary to put these benefits in their proper context.

During each run in the competition, the agent has a set of clients, and each client will pay a fixed amount to obtain an appraisal for one painting. Each painting is classified in one of the finite set of eras. For a given era, the expertise of an agent is modeled by an error distribution. To be more precise, an agent's appraisal is generated by the simulator, from a normal distribution whose mean is the painting's true value and whose deviation is determined by the agent's expertise and the money it spends.[3] Therefore, given that the expertise of the agent is predetermined by the simulator, an agent can only directly affect the accuracy of its appraisals by changing the amount of money it spends to generate the appraisals.

Although the client pays a fixed amount for the appraisals, the client's share allocation strategy depends on the accuracy of such appraisals. This means that low accuracy appraisals can affect the number of clients the agent gets and so, indirectly, its total utility.

Tuning these parameters correctly, we believe, is out of the scope of this work and also out of scope for a trust model. This is the reason why we felt necessary to introduce, as measure of a successful trust model, the metric regarding fulfilled and violated interactions. This also explains why the success of the "social" agents are not proportionally reflected in the total utility.

6. CONCLUSION AND FUTURE WORK

In this article we have outlined a way to combine concepts of social networking and trust relationships. For the first time, we have presented empirical evidence that a technique to build and maintain a social network representation of the environment allows a trust model to be more effective in selecting trustworthy agents. Agents use their social network to obtain knowledge that they could not gather otherwise, and

[3] For more details, see Fullam et al. [2005].

use this knowledge to filter their trust relationships. Although the idea of a social structure has already been presented previously [Sabater and Sierra 2002], there was no indication of how each agent would build this social network representation. The only attempt made is in Ashri et al. [2005]. However, the proposed model has never been implemented or validated.

We have presented a method for agents to build a social network representation of their local environment. Using interactions information such as reputation information, agents can maintain their own representation of such environments. With this extended perception of the environments, agents can make more informed decisions. Results show, on average, an improvement of nearly 3% in the quality of the interactions, over a total of more than a million interactions. With this approach, we strive towards building an archetypal model for trust by combining the concepts of social networking and trust and reputation relationships.

Although the proposed method for identifying social relationships is specific of the particular scenario used for the evaluation, we should consider the general idea behind this. In a general market-like scenario, sellers who sell the same products or provide the same kind of service compete for the same market share, therefore they can be considered in competition, as well as agents needing a particular product or service, to operate in the market, need the "cooperation" or those agents selling that particular product or providing that service.

There is much scope for future work. Further experiments are required to better validate our approach.

First of all, our quantitative results and conclusions are based on simple observations: in Figures 5, 6, and 7, for instance, the results for the social agent simply dominate those for the nonsocial agent under each scenario. For the other figures, more subtle conclusions were drawn. But in all, our results set the scene for a more systematic analysis using an in-depth statistical toolbox on the results for the various scenarios, and it indicates which parameters are interesting to analyze further.

It would be interesting to test the behavior of the "social" agents in presence of agent Uno, and analyze the differences. Furthermore, it would be interesting to explore the possibility of considering each era as a separate market domain, in order to allow the agents to have a more refined view of the social relationships linking each other. Some agents could be in competition for a particular era but in need of each other for another one. Similarly, the information from the social structure could also be used by the agents to change their behavior according to their current goal, whether it is gathering information or providing it.

Moreover, we plan to test our work on different datasets to validate further the general nature of the approach. Having considered the remarks made on the Agent ART testbed, we plan to set up a simpler testbed, where the factors involved in the success of an interaction are only the ones related with the trust decisions. Similar test sets have been used for the well-known Prisoner's Dilemma Competition by Axelrod [1984] and by the more recent Beaufils et al. [1998].

We also plan to refine the way the social structure is updated every time new information is acquired, using link prediction techniques [Liben-Nowell and Kleinberg 2003] and path analysis [Carrington 2005] and explore other measures of similarity.

Finally, we see our work as being influenced by but also contributing to further development of a theoretical framework for social structures in multiagent systems. We have indicated in the 1 how models of social structure are beginning to be considered in the MAS community, and our work gives some hints as to how they can be implemented and validated. We think that in the long term this will help in further advancing the theory of social structures in MAS.

ACKNOWLEDGMENTS

We thank the referees for their useful comments and suggestions.

REFERENCES

ABDUL-RAHMAN, A. AND HAILES, S. 2000. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.*

ASHRI, R., RAMCHURN, S. D., SABATER, J., LUCK, M., AND JENNINGS, N. R. 2005. Trust evaluation through relationship analysis. In *Proceedings of the 4^{th} International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS '05).* ACM Press, New York, 1005–1011.

AXELROD, R. 1984. *The Evolution of Cooperation*. Basic Books.

BEAUFILS, B., DELAHAYE, J.-P., AND MATHIEU, P. 1998. Complete classes of strategies for the classical iterated prisoner's dilemma. In *Evolutionary Programming*, 33–41.

BUSKENS, V. AND WEESIE, J. 2000. Cooperation via social networks. *Analyse und Kritik 22*, 44–74.

CARRINGTON, P. J. AND SCOTTT, J. S.W. 2005. *Models and Methods in Social Network Analysis (Structural Analysis in the Social Sciences).* Cambridge University Press.

CASTELFRANCHI, C. AND FALCONE, R. 1998. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification*, 72–79.

FLACHE, A. AND HEGSELMANN, R. 1999. Rationality vs. learning in the evolution of solidarity networks: A theoretical comparison. *Comput. Math. Organ. Theory 5*, 97–127.

FULLAM, K. K., KLOS, T. B., MULLER, G., SABATER-MIR, J., TOPOL, Z., BARBER, K. S., ROSENSCHEIN, J., AND VERCOUTER, L. 2005. The agent reputation and trust (art) testbed architecture. In *Proceedings of the Conference on Artificial Intelligence Research and Development.* IOS Press, 389–396.

GALASKIEWICZ, J. AND WASSERMAN, S. 1994. *Advances in Social Network Analysis: Research in the Social and Behavioral Sciences*. Sage Publications, Thousand Oaks, CA.

GRANOVETTER, M. 1985. Economic action and social structure: The problem of embeddedness. *Amer. J. Sociol. 91*, 481–510.

HUYNH, T. D., JENNINGS, N. R., AND SHADBOLT, N. R. 2006. An integrated trust and reputation model for open multi-agent systems. *Auton. Agents Multi-Agent Syst. 13*, 2, 119–154.

JURCA, R. AND FALTINGS, B. 2002. Towards incentive-compatible reputation management. In *Trust, Reputation and Security: Theories and Practice,* Lecture Notes in Artificial Intelligence, vol. 2631, Springer, 138–147.

KRUPA, Y., HUBNER, J., AND VERCOUTER, L. 2009. Extending the comparison efficiency of the ART testbed. http://yann.krupa.free.fr/pdf/icore09_khv.pdf.

LIBEN-NOWELL, D. AND KLEINBERG, J. 2003. The link prediction problem for social networks. In *Proceedings of the 12^{th} International Conference on Information and Knowledge Management (CIKM'03).* ACM, New York, 556–559.

MACY, M. W. AND SKVORETZ, J. 1998. The evolution of trust and cooperation between strangers: A computational model. *Amer. Sociol. Rev. 63*, 638–660.

MARSH, S. P. 1994. Formalising trust as a computational concept. http://www.cs.stir.ac.uk/research/publications/techreps/pdf/TR133.pdf.

MULLER, G. AND VERCOUTER, L. 2004. Decentralized monitoring of agent communications with a reputation model. In *Trusting Agents for Trusting Electronic Societies*, 144–161.

MURILLO, J. AND MUNOZ, V. 2008. Agent uno: Winner in the 2nd Spanish art competition. *Inteligencia Artif. Revista Iberoamericana de Inteligencia Artif. 12*, 19–27.

PEARL, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Francisco, CA.

QUATTROCIOCCHI, W., PAOLUCCI, M., AND CONTE, R. 2008. Reputation and uncertainty reduction: Simulating partner selection. In *Trust in Agent Societies*, R. Falcone, S. Barber, J. Sabater-Mir, and M. Singh,Eds, Springer.

RUBIERA, J. C., LOPEZ, J. M. M., AND MURO, J. D. 2001. A fuzzy model of reputation in multi-agent systems. In *Agents*, 25–26.

SABATER, J. AND SIERRA, C. 2002. Reputation and social network analysis in multi-agent systems. In *Proceedings of the International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS'02).* ACM Press, New York, 475–482.

SCHILLO, M., FUNK, P., STADTWALD, I., AND ROVATSOS, M. 2000. Using trust for detecting deceitful agents in artificial societies. http://homepages.inf.ed.ac.uk/mrovatso/papers/schilloetal-aaij2000.pdf.

TAKCS, K., JANKY, B., AND FLACHE, A. 2008. Collective action and network change. *Social Netw. 30,* 177–189.

TEACY, W. T., PATEL, J., JENNINGS, N. R., AND LUCK, M. 2006. Travos: Trust and reputation in the context of inaccurate information sources. *Auton. Agents Multi-Agent Syst. 12*, 2, 183–198.

WALLIS, W. 2007. *A Beginner's Guide to Graph Theory*. Springer.

WEESIE, J., BUSKENS, V., AND RAUB, W. 1998. The management of trust relations via institutional and structural embeddedness. In *The Problem of Solidarity: Theory and Models*, 113–138.

YU, B. AND SINGH, M. P. 2002. Distributed reputation management for electronic commerce. *Comput. Intell. 18*, 4, 535–549.

ZACHARIA, G., MOUKAS, A., AND MAES, P. 1999. Collaborative reputation mechanisms in electronic marketplaces. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences (HICSS '99).* Vol. 8, IEEE Computer Society.

Received August 2010; revised March 2011; accepted March 2011