FIRE: An Integrated Trust and Reputation Model for Open Multi-Agent Systems

T. Dong Huynh and **Nicholas R. Jennings** and **Nigel R. Shadbolt**[1]

Abstract. Trust and reputation are central to effective interactions in open multi-agent systems in which agents, that are owned by a variety of stakeholders, can enter and leave the system at any time. This openness means existing trust and reputation models cannot readily be used. To this end, we present FIRE, a trust and reputation model that integrates a number of information sources to produce a comprehensive assessment of an agent's likely performance. Specifically, FIRE incorporates interaction trust, role-based trust, witness reputation, and certified reputation to provide a trust metric in virtually all circumstances. FIRE is empirically benchmarked and is shown to help agents effectively select appropriate interaction partners.

1 INTRODUCTION

A wide variety of networked computer systems (such as the Grid, the Semantic Web, and peer-to-peer systems) can be viewed as multi-agent systems (MAS) in which the individual components act in an autonomous and flexible manner in order to achieve their objectives [7]. An important class of these systems are those that are *open*; here defined as systems in which agents can freely join and leave at any time and where the agents are owned by various stakeholders with different aims and objectives. From these two features, it can be assured that in open MAS: (1) the agents are likely to be unreliable and self interested; (2) no agent can know everything about its environment; and (3) no central authority can control all the agents.

Despite these many uncertainties, a key component of such systems is the interactions that necessarily have to take place between the agents. Moreover, as the individuals only have incomplete knowledge about their environment and their peers, *trust* plays a central role in facilitating these interactions [8]. Specifically, trust is here defined as a measurable level of the subjective probability with which an agent a assesses that another agent b will perform a particular action, both before a can monitor such action and in a context in which it affects its own action (adapted from [4]). Generally speaking, trust can arise from two views: the individual and the society level. The former consists of agent a's direct experiences from interactions with agent b and the various relationships that may exist between them (e.g. owned by the same organisation, relationships derived from relationships between the agents' owners in the real life such as friendship or relatives). The latter consists of observations by the society of agent b's past behaviour (here termed its *reputation*). These indirect observations are aggregated to define agent b's past behaviour based on the experiences of all the participants in the system.

Given its importance, a number of computational models of trust and reputation have been developed (see Section 4), but none of them are well suited to open MAS. Specifically, given the above characteristics, in order to work efficiently in an open MAS, a trust model needs to possess the following properties:

1. It should take into account a variety of sources of trust information in order to have a more precise trust measure (by cross correlating several perspectives) and to cope with the situation that some of the sources may not be available.
2. Each agent should be able to evaluate trust for itself. Given the 'no central authority' nature of an open MAS, agents will typically be unwilling to rely solely on a single centralised trust/reputation service.
3. It should be robust against possible lying from agents (since the agents are self-interested).

To deal with these requirements, we developed a new trust and reputation model called FIRE[2]. In so doing, we advance the state of the art in the following ways. We developed a modular model that integrates four different types of trust and reputation: *interaction trust* (resulting from past experiences from direct interactions), *role-based trust* (defined by various role-based relationships between the agents), *witness reputation* (reports of witnesses about an agent's behaviour), and *certified reputation* (references provided by other agents about its behaviour). This breadth is important in our domain because it enables an agent to combine a variety of alternative sources of information (to cope with the inherent uncertainties) and because in various circumstances not all of these sources will be readily available (but a measure of trust is nevertheless needed to interact).

Of particular relevance is the introduction of a novel type of reputation — certified reputation. The other more traditional ways of building a trust measure (i.e. interaction and role-based trust and witness reputation) have certain limitations. For example, if agent a has not interacted with b before, it has no information to calculate its interaction trust. In the case of witness reputation, a may not be able to find relevant witness ratings about b, or the search process may take too long to finish. Finally, there may be no role-base relationships with b. If all these things happen at the same time (e.g. agent a has just joined the environment), agent a will not be able to assess agent b's trustworthiness. In such situations, if agent b can present certified information about its past performance to a (in the form of references from other agents who have interacted with it), agent a will then be able to make some assessment of its trustworthiness.

The remainder of the paper is organised as follows. In the next section, we will present the FIRE model and its components. The model will then be empirically evaluated in Section 3. Section 4 presents

[1] School of Electronics and Computer Science, University of Southampton, UK. Emails: {tdh02r,nrj,nrs}@ecs.soton.ac.uk.

[2] FIRE is from '**fi**des' (Latin for 'trust') and '**re**putation'. In the Ramayana legend of India, Sita proved the purity of her character by passing through the raging fire flames.

related work in the area. Finally, Section 5 concludes this paper and outlines the future work.

2 THE FIRE MODEL

FIRE is an integrated trust and reputation model consisting of four main components: interaction trust, role-based trust, witness reputation, and certified reputation. Each of these components will be presented in turn and Section 2.5 will then show how these components are combined together.

2.1 Interaction trust

Interaction trust (IT) models the trust that ensues from the direct interactions between two agents. Here we simply exploit the direct trust component of Regret [9] since this meets all our requirements for dealing with direct experiences. In more detail, consider a commercial transaction where agent a buys a particular product from agent b. The outcome of the transaction may consist of the product price, product quality, and the delivery date. From this outcome, agent a may give ratings about agent b's service in terms of price, quality, and delivery for that particular interaction. Ratings are thus tuples in the following form: $r = (a, b, i, c, v)$, where a and b are the agents that participated in the interaction i, and v is the rating a gave b for the term c (e.g. price, quality, delivery). The range of v is $[-1, +1]$, where -1 means absolutely negative, $+1$ means absolutely positive, and 0 means neutral or uncertain.

In order to calculate IT from past experiences, an agent needs to record its past ratings in a (local) *rating database*. When calculating the IT value for agent b with respect to term c, agent a has to query its database for all the ratings that have the form $(a, b, \_, c, \_)$, where the '_' symbol can be replaced by any value. We call the set of those ratings $\mathcal{R}(a, b, c)$. Then the IT (denoted by $\mathcal{T}_I$) is calculated as the weighted mean of the rating values of all the ratings in the set:

$$\mathcal{T}_I(a, b, c) = \sum_{r_i \in \mathcal{R}(a,b,c)} \omega(r_i) \cdot v_i, \tag{1}$$

where v_i is the value of the rating r_i and $\omega(r_i)$ is the weight corresponding to r_i. The weight $\omega(r_i)$ for each rating is selected such that it gives more weight to more recent ratings, with a constraint that $\sum_{r_i \in \mathcal{R}(a,b,c)} \omega(r_i) = 1$. This is to ensure that the trust value $\mathcal{T}_I(a, b, c)$ is in the range $[-1, +1]$.

In FIRE, each trust value comes with a reliability rating that reflects the confidence of the trust model in producing that trust value given the data it took into account. This value is built from the two following measures:

- $\rho_N(a, b, c)$: the reliability measure based on the number of ratings that have been taken into account in computing $\mathcal{T}_I$. As the number of these ratings (n) grows, the degree of reliability increases until it reaches a defined threshold (denoted by m).

$$\rho_N(a, b, c) = \begin{cases} \frac{n}{m} & \text{when } n \leq m \\ 1 & \text{when } n > m \end{cases}, \tag{2}$$

  where n is the cardinality of the set $\mathcal{R}(a, b, c)$. The value of function $\frac{n}{m}$ ranges from 0 to 1 for n in $[0, m]$. Hence, the reliability $\rho_N(a, b, c)$ increases from 0 to 1 when the number of ratings n increases from 0 to m, and stays at 1 when n exceeds m.
- $\rho_D(a, b, c)$: the rating deviation reliability. The greater the variability in the rating values, the more volatile the other agent will be in fulfilling its agreements:

$$\rho_D(a, b, c) = 1 - \sum_{r_i \in \mathcal{R}(a,b,c)} \frac{\omega(r_i) \cdot |v_i - \mathcal{T}_I(a, b, c)|}{2}, \tag{3}$$

Then, the reliability measure of IT (called $\rho_{\mathcal{T}_I}(a, b, c)$) is defined by the following formula:

$$\rho_{\mathcal{T}_I}(a, b, c) = \rho_N(a, b, c) \cdot \rho_D(a, b, c) \tag{4}$$

2.2 Role-based trust

Role-based trust (RT) models the trust resulting from the role-based relationships between two agents (e.g. owned by the same organisation, relationships derived from relationships between the agents' owners in the real life such as friendship or relatives). Since there is no general method for computationally quantifying trust based on this type of relationship, we use rules to assign RT values. This means end users can add new rules to customise this component to suit their particular applications. Rules are tuples of the following form: $rul = (role_a, role_b, c, v_D, e_D)$, which describes a rule that if $role_a$ and $role_b$ are the roles of agent a and b respectively, then the expected performance of b in an interaction with a is v_D ($v_D \in [-1, 1]$) with respect to the term c; $e_D \in [0, 1]$ is the default level of influence of this rule on the resulting RT value. For example, possible rules may be:

$rul_1 = (\text{buyer}, \text{seller}, \text{quality}, -0.2, 0.3)$,
$rul_2 = (\text{friend-buyer}, \text{friend-seller}, \text{quality}, 0, 0.6)$, and
$rul_3 = (\_, \text{government-seller}, \text{quality}, 0, 0.9)$.

rul_1 expresses an agent's belief that an ordinary seller will usually sell a product of slightly lower quality than agreed, but the reliability of this belief is low (0.3); rul_2 is the belief that in a close partnership the buying agent can expect the seller to do what is agreed in terms of product quality; and this is also true for a governmental seller almost all of the time (rul_3).

Each agent has its own set of rules which are stored in a (local) rule database. In order to determine the RT with an agent b, agent a looks up the relevant rules from its rule database. Then the value of RT is given by the following formula:

$$\mathcal{T}_R(a, b, c) = \frac{\sum_{rul_i \in Rules(a,b,c)} e_{Di} \cdot v_{Di}}{\sum_{rul_i \in Rules(a,b,c)} e_{Di}}, \tag{5}$$

where $rul_i = (role_a, role_b, c, v_{Di}, e_{Di})$ is a rule in the set of rules $Rules(a, b, c)$. This set is a subset of the rule database in which only the rules that are relevant to the roles of a, the roles of b, and the term c are selected.

Since the rules for RT are specified by the agent's owner, the reliability of RT also needs to be set by the agent's owner. We use $\rho_{\mathcal{T}_R}(a, b, c)$, which ranges in $[0, 1]$, to denote this value.

2.3 Witness reputation

The *witness reputation* (WR) of a target agent b is built on observations about its behaviour by other agents (witnesses). In order to evaluate the WR of b, an agent a needs to find the witnesses that have interacted with b. In this component, we use a variant of the referral system in [11] to find such witnesses. In our system, agents cooperate by giving, pursuing, and evaluating referrals (a recommendation to contact another agent). Each agent in the system maintains a list of acquaintances (other agents that it knows). Thus, when looking for a certain piece of information, an agent can send the query to a number of its acquaintances who will try to answer the query if possible or, if they cannot, they will send back referrals pointing to other agents that they believe are likely to have the desired information.

In this model, each agent has a measure of the degree of likeliness with which an agent can fulfil an information query. This measure needs to be defined in an application specific manner. For example, in our testbed (described in Section 3.1), an agent is assumed to know local agents (those who are near to it) better and so we use the distance between an acquaintance and the target agent as the knowledge measure. Thus the nearer to the target agent, the more likely the acquaintance is to know it. When an agent a assesses the WR of an agent b with respect to a term c, denoted by $\mathcal{T}_W(a,b,c)$, it sends out a query for ratings of the form $(\_, b, \_, c, \_)$ to those acquaintances that are likely to have relevant ratings on agent b and term c. These acquaintances, upon receiving the query, try to match it to their own rating databases. If they find matching ratings, it means they have had interactions with b, and they will return the ratings found to a. If they cannot find the requested information, they will return referrals identifying their acquaintances that they believe are most likely to have the relevant ratings to the query so that a can look further. This process continues until a finds sufficient witnesses or the lengths of its referral chains reach a defined threshold (because the further the witness is from a, the less reliable/relevant its information is to it). The general formula for WR is as follows:

$$\mathcal{T}_W(a,b,c) = \sum_{r_i \in \mathcal{R}_W(a,b,c)} \omega(r_i) \cdot v_i \qquad (6)$$

where $\mathcal{R}_W(a,b,c)$ is the set of witness ratings found by agent a, the weight $\omega(r_i)$ for each rating is defined as per Section 2.1, and v_i is the rating value of r_i. The reliability measure for WR (denoted by $\rho_{\mathcal{T}_W}(a,b,c)$) is also defined from the ratings in $\mathcal{R}_W(a,b,c)$ as per Section 2.1.

2.4 Certified reputation

Certified reputation (CR) are ratings presented by the rated agent (agent b) about itself which have been obtained from its partners in past interactions [6]. These ratings are certifications (provided by the rating agents) of agent b's past performance (somewhat like a reference when applying for a job). They allow an agent to prove its achievable performance as viewed by previous interaction partners[3]. Since agent b can choose which ratings it puts forward, a rational agent will only present its best ratings. Therefore, we should assume that CR information possibly overestimates an agent's expected behaviour. Thus, although it cannot guarantee agent b's performance in future interactions, the CR information does reveal a partial perspective on agent b's past behaviour. The main benefit of this type of information is its high availability. With the cooperation of its partners, agent b can have CR information from just a small number of interactions. Therefore, CR is available to agents in most circumstances; even in situations where the other components may fail to provide a trust measure. In more detail, the process of CR is as follows:

- After every transaction, b asks its partners to provide their ratings about its performance which it stores in its databases.
- When a contacts b to express its interest in using b's service it asks b to provide references about its past performance.
- Agent a receives the ratings of b from b. It assesses the ratings' reliability and calculates a trust value for b. Specifically, the value of CR, $\mathcal{T}_C(a,b,c)$, and its reliability, $\rho_{\mathcal{T}_C}(a,b,c)$, are calculated as per the WR component, but the input is the set of ratings provided by the target agent b itself.

2.5 Combining the components

We combine the aforementioned trust/reputation values into a single composite measure to give an overall picture of an agent's likely performance. Specifically, we use the weighted mean method to calculate the composite trust value ($\mathcal{T}(a,b,c)$) and its reliability ($\rho_{\mathcal{T}}(a,b,c)$):

$$\mathcal{T}(a,b,c) = \frac{\sum_{k\in\{I,R,W,C\}} w_k \cdot \mathcal{T}_k(a,b,c)}{\sum_{k\in\{I,R,W,C\}} w_k} \qquad (7)$$

$$\rho_{\mathcal{T}}(a,b,c) = \frac{\sum_{k\in\{I,R,W,C\}} w_k}{\sum_{k\in\{I,R,W,C\}} W_k} \qquad (8)$$

where $w_k = W_k \cdot \rho_{\mathcal{T}_k}(a,b,c)$, and W_I, W_R, W_W, W_C are the coefficients corresponding to the IT, RT, WR, and CR components. These coefficients are set by end users to reflect the importance of each component in a particular application.

3 EMPIRICAL EVALUATION

3.1 The testbed

The testbed environment for evaluating FIRE is a multi-agent system consisting of agents providing services (called *providers*) and agents using those services (called *consumers*). Without loss of generality, it is assumed that there is only one type of service in the testbed. Hence, all the provider agents offer the same service. However, their performance (i.e. the quality of the service) may be different. The agents are situated randomly on a spherical world (see Figure 1) and are stationary through their life cycles. Each agent has a *radius of operation* (r_o – depicted by a dotted circle around an agent in Figure 1) that models the agent's capability in interacting with others (e.g. the available bandwidth or the agent's infrastructure) and any agents situated in that range are the agent's neighbours.

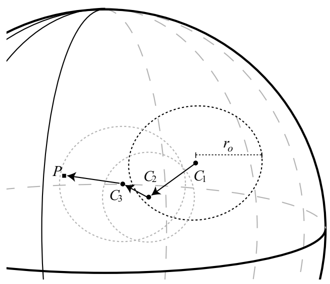

Figure 1. The spherical world and a path from consumer C_1 (through C_2 and C_3) to provider P based on neighbourhood.

Simulations are run in the testbed in rounds (of agent interactions). In each round, if a consumer agent needs to use the service it can contact the environment to locate nearby provider agents (in terms of the distance between the agents on the spherical world). The consumer agent will then select one provider from the list to use its service. The selection process relies on the agent's trust model to decide which provider has the most reliable service. Consumer agents without a trust model randomly select a provider from the list. The consumer agent then uses the service of the selected provider and gains some utility from the interaction (called UG). The value of UG is in $[-10, 10]$ and depends on the level of performance of the provider in that interaction. A provider agent can serve many users at a time.

After an interaction, the consumer agent will rate the service of the provider based on the level of performance it received. It records the rating for subsequent trust evaluations and also informs the provider about the rating it made. The provider may record the rating as evi-

[3] It is assumed that some form of security mechanism (such as a public-key infrastructure) is employed to ensure that the provided references cannot be tampered with.

dence about its performance to be presented to potential consumers[4]. In order to simulate the openness of the environment, after each round of simulation, a (random) number of consumers leave the testbed, and a (random) number of new consumers are added.

In our testbed the only difference in each situation is the performance of the provider agents. We consider four types of provider agents: good, ordinary, bad, and intermittent. Each of them, except the last, has a mean level of performance (μ_{P}). Its actual performance follows the normal distribution around this mean. The values of μ_{P} and the associated standard deviation (σ_{P}) of these types of providers are given in Table 1. Intermittent providers, on the other hand, yield unpredictable (random) performance levels in the range $[\mathsf{PL\_BAD}, \mathsf{PL\_GOOD}]$. In addition, the service quality of a provider is also degraded linearly in proportion to the distance between the provider and the consumer to reflect the greater uncertainties associated with service delivery.

Profile	Range of μ_{P}	σ_{P}
Good	[PL_GOOD, PL_PERFECT]	1.0
Ordinary	[PL_OK, PL_GOOD]	2.0
Bad	[PL_WORST, PL_OK]	2.0

Performance level	Utility gained
PL_PERFECT	10
PL_GOOD	5
PL_OK	0
PL_BAD	-5
PL_WORST	-10

Table 1. Profiles of provider agents.

3.2 Experimental methodology

In each experiment, the testbed is populated with provider and consumer agents. Each consumer agent is equipped with a particular trust model, which helps it select a provider when it needs to use a service. Since the only difference among consumer agents is the trust models that they use, the utility gained by each agent through simulations will reflect the performance of its trust model in selecting reliable providers for interactions. Therefore, the testbed records the UG of each interaction along with the trust model used. In order to obtain an accurate result for performance comparisons between trust models, each one will be employed by a large number of consumer agents ($\mathsf{N_C}$). In addition, the UG means of agents employing the same trust models (called consumer groups) are compared with each other's using the two-sample t-test [2] (for means comparison) with the confidence level of 95%. The result of an experiment is then presented in a graph with two y-axes; the first plots the UG means of consumer groups in each interaction and the second plots the corresponding performance rankings obtained from the t-test (prefixed by R., where the group of rank 2 outperforms that of rank 1). The experimental variables for the experiments are presented in Table 2 and their values will be used in all cases unless otherwise specified.

Simulation variable	Symbol	Value
Number of simulation rounds	N	500
Total number of provider agents:	$\mathsf{N_P}$	100
+ Good providers	$\mathsf{N_{PG}}$	10
+ Ordinary providers	$\mathsf{N_{PO}}$	55
+ Intermittent providers	$\mathsf{N_{PI}}$	25
+ Bad providers	$\mathsf{N_{PB}}$	10
Number of consumer agents in each group	$\mathsf{N_C}$	500
Maximum number of new consumer agents joining each round	$\mathsf{N_{CJ}}$	10
Maximum number of old consumer agents leaving each round	$\mathsf{N_{CL}}$	10

Table 2. Experimental variables.

[4] It is assumed that all agents are honest in exchanging information in this testbed. The problem of strategic behaviour in reporting this information will be considered in future work.

3.3 Overall performance of FIRE

In order to evaluate the overall performance of FIRE, we compare it with the SPORAS model[5] (whose operation is described in Section 4) and a group of agents with no trust model. Hence, there are three groups of consumer agents: FIRE, SPORAS, and NoTrust. As can

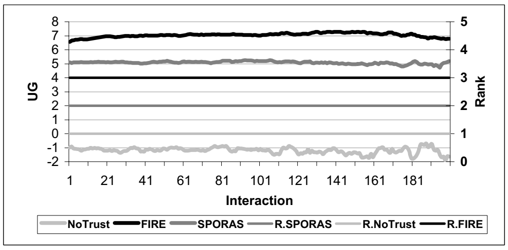

Figure 2. Comparing FIRE with SPORAS and the no-trust case.

be seen from Figure 2, the NoTrust group, selecting providers randomly without any trust evaluation, performs consistently the lowest. FIRE, however, manages to gain about 40% more utility than SPORAS, the second rank, throughout the interactions. This gain is accounted for by the fact that FIRE separates direct experiences from others' experiences (i.e. ratings) in trust evaluation, while SPORAS treats all types of ratings equally. Therefore, SPORAS suffers from noise in ratings (resulting from different degrees of degradation of service quality due to different provider-consumer distances). In contrast, FIRE reduces rating noise by giving more weights to direct experiences[6], which are more relevant to an individual agent's situation.

3.4 Performance of FIRE's components

We argued that each component of FIRE plays an important role in exploiting trust information from a particular source and this, in turn, contributes to the effectiveness of the overall model. In order to confirm this, we benchmark FIRE with and without various components to evaluate the contribution of that component to the whole model. However, since the IT component is reused from Regret, we will only focus on evaluating the novel components (i.e. the WR and CR components). Role-based trust is not considered because it is typically highly domain specific.

First, we benchmark the WR component. In this experiment, there are two groups of consumer agents. The first one uses only the IT component (called the control group). The second makes use of the WR component in addition to the IT component (called the WR group). The result of the experiment, presented in Figure 3, shows that the WR component substantially improves the performance of consumer agents in the first 40 interactions. The t-test ranking also confirms this by showing that agents using the WR component outperform agents using only the IT component in a similar period. In later interactions, both groups perform equally. This can be explained by the fact that witness ratings collected by agents employing the WR component actually help produce better trust values. After 50 interactions, agents evaluating only IT catch up with the other group as they have learned about the environment from past (bad) experiences.

[5] SPORAS is a successful centralised trust model which is often used for benchmarking. Therefore, we choose it so that FIRE can be compared relatively with current trust models, as well as to compare our model with those that follow the centralised approach.

[6] In all our experiments, we set W_I, W_R, W_W, W_C to 2.0, 4.0, 1.0, and 0.5 respectively to reflect the fact that direct experiences are more reliable than those from witnesses, and CR information from the target agent itself is the least reliable.

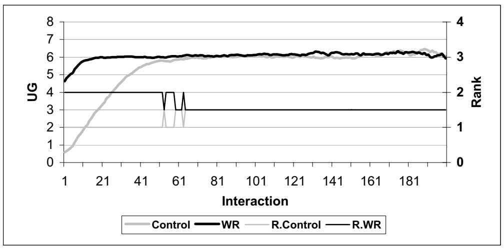

Figure 3. Performance of the WR component.

In the next experiment we evaluate the CR component (using a similar setting). Here there are two groups of consumer agents. The control group employs the IT and WR components, and the other employs the CR component in addition (called the CR group). Figure 4

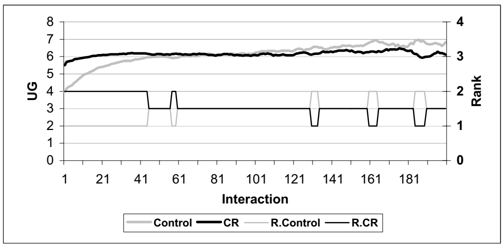

Figure 4. Performance of the CR component.

shows that, by employing the CR component, the CR group outperforms the control group by 1.5 utility units on the very first interaction. The CR group then manages to maintain its lead for the first 44 interactions before the control group catches up (as confirmed by the t-test). This improvement shows that CR information from the providers helps FIRE to produce a more precise trust measure in the first few interactions, whereas the IT and WR components perform inefficiently due to the scarcity of trust information. In later interactions, the CR group performs in a comparable way to the control group. However, there are a small number of times it falls behind, though the gap is small. This can be explained by the fact that, in the CR component, the providers always give the consumers their best ratings and these do not always reflect their actual performance. This suggests the importance of the CR component should be dynamically adjusted at different stages of an agent's life cycle.

4 RELATED WORK

Probably the most widely used reputation models are those on eBay [3] and Amazon Auctions [1]. Both of these are implemented as a centralised rating system so that their users can rate and learn about each other's reputation. For example, an eBay user, after an interaction, can rate its partner on a scale of -1, 0, or $+1$. The ratings are stored centrally and the reputation value is computed as the sum of those ratings over six months. Thus, reputation in these models is a global single value. However, these models are too simple (in terms of their trust rating values and the way they are aggregated) for applications in open MAS.

SPORAS [12] extends these models by introducing a new method for rating aggregation. Specifically, it does not store all the ratings, but rather updates the global reputation value of an agent according to its most recent rating. In addition, it introduces a reliability measure based on the standard deviations of the rating values. However, treating all ratings equally means SPORAS suffers from rating noise (as shown in Section 3) and its centralised approach is not suitable for our target domain.

Regret decentralises the trust evaluation process and each agent stores its ratings in its local database (see Section 2.1). This enables the model to introduce a more realistic trust measure from ratings of richer semantics and to give more weight to recent ratings. Regret also presents a witness reputation component along with a sophisticated method for aggregating witness reports. However, it does not show how witnesses can be located, and, thus, this component is not of much use. We overcome this in FIRE by employing a referral process in which agents help each other to find witnesses based on their expertise (see Section 2.3).

The certified reputation component in FIRE has similarities to trust policy management engines such as PolicyMaker [5] and TrustServ [10]. These engines grant rights to an agent based on its certificates according to defined policies (i.e. rules). In contrast, our CR component computationally evaluates information provided by an agent to deduce its trustworthiness for selecting interaction partners.

5 CONCLUSIONS AND FUTURE WORK

This paper has presented a novel decentralised model for trust evaluation in open MAS in which each agent is responsible for storing trust information and evaluating trust itself. Through empirical evaluation, we showed how FIRE helps agents to select more reliable partners for interaction and thus obtain better utility. The main benefit of FIRE is that it can produce a trust measure and an associated reliability measure in most situations. Moreover, with its generic design, FIRE can be easily adapted to various domains because of its modularised design and parameterised configuration. In short, it satisfies the first two requirements for a trust model in open MAS as specified in Section 1. However, at present, it assumes the agents report their trust information truthfully. As noted in the requirements, this is not suitable for our target domain and, for this reason, we plan to devise reliability measures for witness ratings and certified ratings that take into account the possibility of lying. This will make the model more robust and ready to be used in real open MAS applications.

REFERENCES

[1] Amazon Site. `http://www.amazon.com`. World Wide Web.
[2] P. R. Cohen, *Empirical Methods for Artificial Intelligence*, The MIT Press, 1995.
[3] eBay Site. `http://www.ebay.com`. World Wide Web.
[4] D. Gambetta, 'Can we trust trust?', in *Trust: Making and Breaking Cooperative Relations*, ed., Diego Gambetta, 213–237, Department of Sociology, University of Oxford, electronic edn., (2000).
[5] T. Grandison and M. Sloman, 'A survey of trust in internet applications', *IEEE Communications Surveys & Tutorials*, **3**(4), (2000).
[6] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, 'Developing an integrated trust and reputation model for open multi-agent systems', in *Proc. 7th Int Workshop on Trust in Agent Societies*, (2004).
[7] N. R. Jennings, 'An agent-based approach for building complex software systems', *Communications of the ACM*, **44**(4), 35–41, (2001).
[8] S. D. Ramchurn, T. D. Huynh, and N. R. Jennings, 'Trust in multi-agent systems', *The Knowledge Engineering Review*, (2004).
[9] J. Sabater, *Trust and Reputation for Agent Societies*, Phd thesis, Universitat Autnoma de Barcelona, 2003.
[10] H. Skogsrud, B. Benatallah, and F. Casati, 'Model-driven trust negotiation for web services', *IEEE Internet Computing*, **7**(6), 45–52, (2003).
[11] B. Yu and M. P. Singh, 'Searching social networks', in *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. ACM Press, (2003).
[12] G. Zacharia and P. Maes, 'Trust management through reputation mechanisms', *Applied Artificial Intelligence*, **14**(9), 881–908, (2000).