Developing an Integrated Trust and Reputation Model for Open Multi-Agent Systems

Dong Huynh, Nicholas R. Jennings, Nigel R. Shadbolt
School of Electronics and Computer Science
University of Southampton, UK.
{*tdh02r,nrj,nrs*}*@ecs.soton.ac.uk.*

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 most 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 assumed 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 [9, 4]. Specifically, trust is here defined as 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, relationships between a service provider agent and its registered consumer agents). The latter consists of observations by the society of agent b's past behaviour (here termed its *reputation*). These indirect observations are aggregated in some way 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[1]. In so doing, we ad-

1 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

vance 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 the agent's 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 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 to provide a single measure.

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 [10] 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*. This database stores at maximum the H latest ratings that the agent gave to the each partner that it has interacted with. Here H is called the *local rating history size*. 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 $\mathcal{R}_{\mathsf{L}}(a, b, c)$. 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:

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

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 $\sum_{r_i \in \mathcal{R}_{\mathsf{L}}(a,b,c)} \omega(r_i) = 1$. 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 (called the *rating intimacy threshold*, and denoted by m).

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

  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.

passing through the raging fire flames.

- $\rho_D(a,b,c)$: the rating deviation reliability. The greater the variability in the rating values, the more volatile the other agent is likely to be in fulfilling its agreements:

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

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

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

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 company, a service provider and its registered user, friendship relationship of their owners). 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:

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

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 $\rho_{\mathcal{T}_R}(a,b,c)$, again in the range $[0,1]$, to denote this value. A reliability measure that can be used for role-based trust is:

$$\rho_{\mathcal{T}_R}(a,b,c) = \frac{\sum_{rul_i \in Rules(a,b,c)} e_{Di}}{|Rules(a,b,c)|}, \quad (6)$$

where $|Rules(a,b,c)|$ is the cardinality of the set of rules.

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 [12] 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 n_{BF} (called the *branching factor*) 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 (local) 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 n_{BF} 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 denoted by n_{RL} (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:

$$\mathcal{T}_W(a,b,c) = \sum_{r_i \in \mathcal{R}_W(a,b,c)} \omega(r_i) \cdot v_i \quad (7)$$

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[2]. 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 probably 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. An overall value

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 ($\rho_{\mathcal{T}}(a,b,c)$):

$$\mathcal{T}(a,b,c) = \frac{\sum_{k\in\{I,R,W,C\}} w_k \cdot \mathcal{T}_k(a,b,c)}{\sum_{k\in\{I,R,W,C\}} w_k} \quad (8)$$

$$\rho_{\mathcal{T}}(a,b,c) = \frac{\sum_{k\in\{I,R,W,C\}} w_k}{\sum_{k\in\{I,R,W,C\}} W_k} \quad (9)$$

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

In order to empirically evaluate FIRE, we design a testbed characterising an open MAS (Section 3.1). The methodology used for the evaluation is then described in Section 3.2. The two last sections (Section 3.3 and 3.4) present the experiments and their results. The former shows the overall performance of FIRE, and the latter shows the contributions of each of FIRE's component to its overall performance.

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) differs. The agents are situated randomly on a spherical world whose radius is 1.0 (see Figure 1). 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.

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 is likely to be the most reliable. 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

2 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.

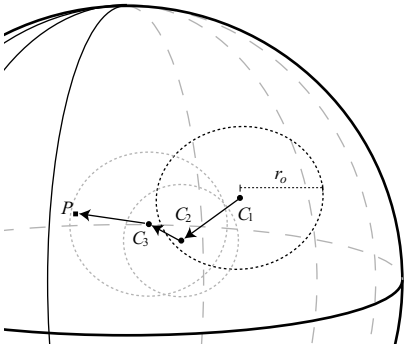


Figure 1. The spherical world and a path from consumer C_1 (through C_2 and C_3) to provider P based on neighbourhood.

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.

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 evidence about its performance to be presented to potential consumers[3].

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 a 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 [PL_BAD, 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 (e.g. increased delays or losses in information exchanges between two agents when they are far away from each other that affect the service quality).

Moreover, in order to simulate different levels of dynamism of an open MAS, a number of factors of the testbed are changed after each round:

- *The population of agents*: In an open MAS, agents can come and leave the system at anytime. This is simulated by removing a number of randomly selected agents from the testbed and adding new agents into it. The numbers of agents added and removed after each round vary, but have an upper limit of some predefined percentage of the whole population. The population change limits for the consumer and the provider populations are denoted respectively by p_{CPC} and p_{PPC}. Since providers are usually more established than consumers, p_{PPC} is set to be lower than p_{CPC} in our simulations. The characteristics of the newly added agents are set randomly but they are uniformly distributed over the the initial agent populations (i.e. the proportions of providers of different profiles and that of consumers in different groups are maintained).
- *The locations of agents*: During their life cycle, agents break old relationships and make new ones (reflecting the notion of continual change that are inherent in open MAS). In our testbed, this type of change is reflected by the change in an agent's location on the spherical world. When a consumer changes its location, it will have a new set of neighbours according to its r_o. In addition, the location of an agent in the testbed also reflects its individual situation covering things such as its knowledge about other local agents (see Section 2.3) and the service delivery between providers and consumers (see above). Therefore, changing an agent's location changes its relationships with others, as well as its individual situation. Specifically, we use the polar coordination (r, φ, θ) for agent locations on the spherical world. Then in order to change an agent's location, amounts of angular changes $\Delta\varphi$ and $\Delta\theta$ are added to φ and θ respectively. $\Delta\varphi$ and $\Delta\theta$ are selected randomly in $[-\Delta\phi, +\Delta\phi]$. Thus, $\Delta\phi$ limits the variability of agents' locations. Not every agent changes its locations every round and, in particular, p_{CLC} and p_{PLC} are used to denote the probabilities that a consumer or a provider respectively changes its location in a round.
- *The behaviour of the providers*: In many environments, provider performance may alter (for better or worse) over time. A provider may even change its behaviour completely (e.g. a provider may take advantage of its

3 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.

good reputation and decide to perform selfishly to obtain better utility). In our testbed, the average performance of a provider (μ) is changed by an amount of $\Delta\mu$ randomly selected in $[-\mathsf{M}, +\mathsf{M}]$, and this happens in each round with the probability of $p_{\mu\mathsf{C}}$. Moreover, after each round, a provider can switch to a completely new provider profile with a probability of $p_{\mathsf{ProfileSwitch}}$.

The values of the variables above are selected to reflect environments that have varying degrees of dynamism (see Table 2) ranging from 0 (completely static) to 3 (highly dynamic).

Level	0	1	2	3
p_{CPC}	0%	1%	2%	5%
p_{PPC}	0%	0%	1%	2%
$p_{\mu\mathsf{C}}$	0%	5%	5%	10%
M	0.0	0.5	1.0	1.0
$p_{\mathsf{ProfileSwitch}}$	0%	1%	2%	4%
p_{CLC}	0%	2%	5%	10%
p_{PLC}	0%	2%	5%	10%
$\Delta\phi$	0	$\frac{\pi}{40}$	$\frac{\pi}{40}$	$\frac{\pi}{20}$

Table 2. Levels of dynamism of the testbed.

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 average UG 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). Each experiment is repeated for each level of dynamism from 0 to 3 to show how dynamism affects the performance of trust models. The experimental variables for the various experiments are presented in Table 3 and their values will be used in all cases unless otherwise specified. The parameters of FIRE that are set for the experiments are also shown in Table 4.

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

Table 3. Experimental variables.

Parameters	Symbol	Value
Local rating history size	H	10
Rating intimacy threshold	m	10
Braching factor	n_{BF}	2
Referal length threshold	n_{RL}	5
Component coefficients:		
+ Interaction trust	W_I	2.0
+ Role-base trust	W_R	4.0
+ Witness reputation	W_W	1.0
+ Cerified reputation	W_C	0.5

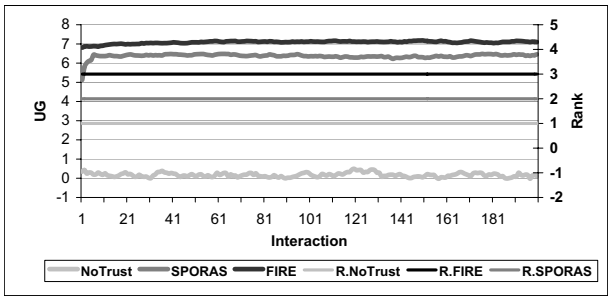
Table 4. FIRE's default parameters.

3.3. Overall performance of FIRE

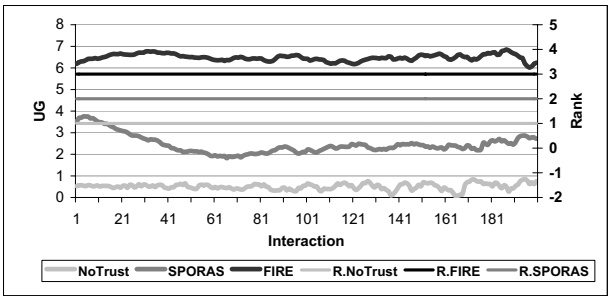
In order to evaluate the overall performance of FIRE, we compare it with the SPORAS model[4] (whose operation is described in Section 4) and a group of agents with no trust model. Hence, there are three groups of consumer agents: FIRE, SPORAS, and NoTrust.

As can be seen from Figure 2, in a static environment (dynamism level 0), the NoTrust group, selecting providers randomly without any trust evaluation, performs consistently the lowest; and FIRE outperforms SPORAS, the second rank, throughout the interactions. It should be noted that SPORAS, being a centralised model, gathers much more

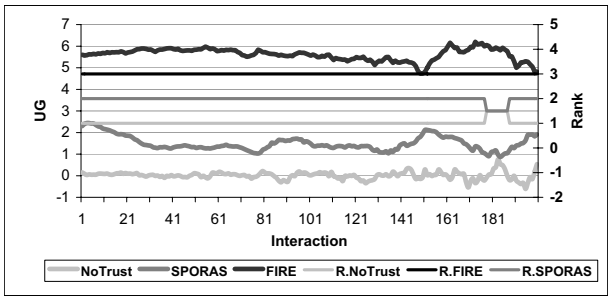
4 SPORAS is a successful centralised trust model which is often used for benchmarking. Therefore, we choose it so that FIRE can be relativised against another trust model, as well as to compare our model with those that follow the centralised approach.



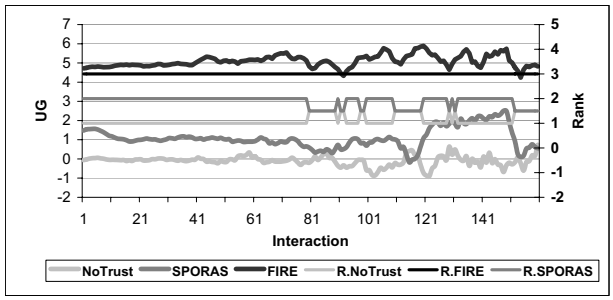
(a) Dynamism level 0



(b) Dynamism level 1

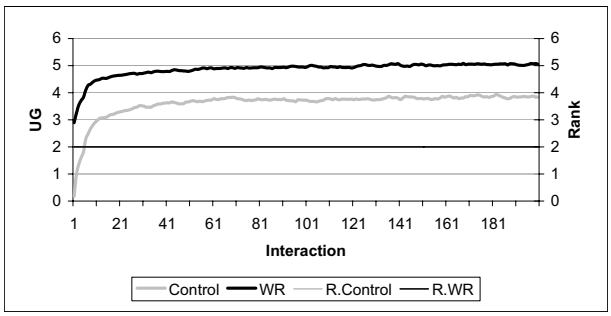


(c) Dynamism level 2

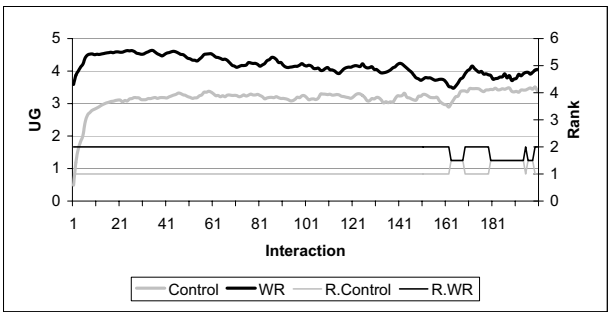


(d) Dynamism level 3

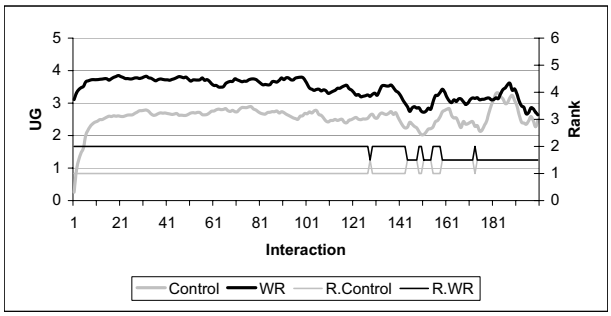
Figure 2. Comparing FIRE with SPORAS and the no-trust case.



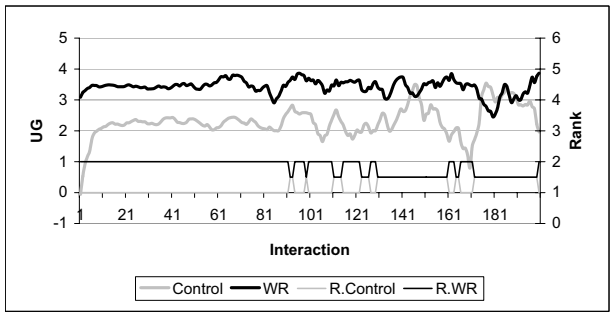
(a) Dynamism level 0



(b) Dynamism level 1

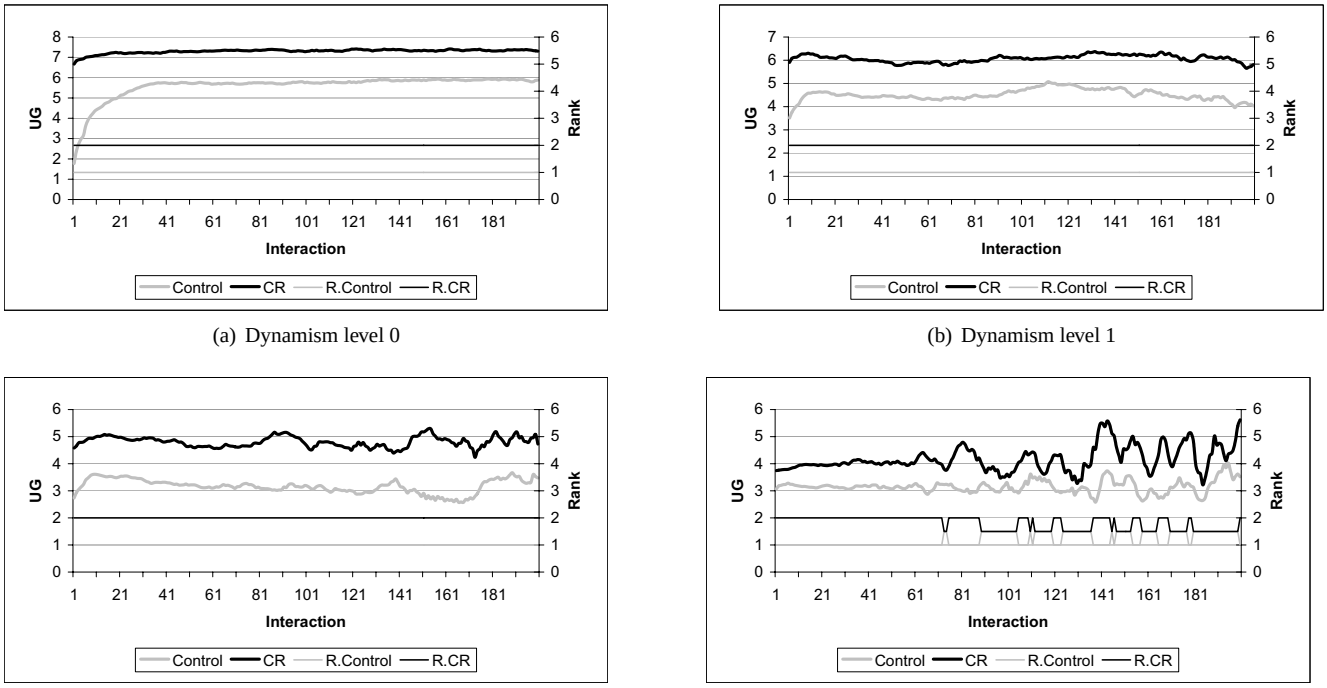


(c) Dynamism level 2



(d) Dynamism level 3

Figure 3. Performance of FIRE's WR component.



(a) Dynamism level 0

(b) Dynamism level 1

(c) Dynamism level 2

(d) Dynamism level 3

Figure 4. Performance of FIRE's CR component.

information than FIRE (a decentralised model)[5]. However, the utility difference of FIRE and SPORAS 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.

As the dynamism of the environment increases (from dynamism level 0 to 1, 2 and 3), the NoTrust group still performs consistently badly (UG around 0.0). However, the dynamism affects FIRE and SPORAS more significantly and their performance decreases as the levels of dynamism increases (as expected). As discussed above, since SPORAS takes all ratings into account and treats them equally, it suffers a lot from the increasing level of noise due to the increasing dynamism. Therefore, its performance decreases dramatically; sometimes being close to that of the NoTrust group (see Figure 2(d)). Although FIRE also suffers from the increasing dynamism, taking the recency of ratings into account (from Equation 1) allows it to quickly adapt to changing situations. In addition, FIRE reduces the adverse effect of noise by taking only relevant ratings into account and weighing the rating sources according to their reliability/relevancy (using reliability measures and predefined coefficients for its components from Equation 8). Thanks to all these features, FIRE manages to maintain a high level of utility gained in the dynamic environments (i.e. an average UG of around 5.0 as compared with that of around 7.0 in the static environment).

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 be-

5 After every interaction, the consumer reports its rating about the provider's service in that interaction to SPORAS. Therefore, SPORAS collects all the available ratings from its users. In contrast, consumers employing FIRE have only ratings from a limited set of witnesses (from the WR component) and those presented by providers (from the CR component) in addition to their own ratings.

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.

cause 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 a static environment. The t-test ranking also confirms this by showing that agents using the WR component outperform agents using only the IT component in all interactions. In more dynamic environments, the performance of WR decreases since witness ratings are out of date due to changes in the environments. However, it always maintains a higher, or at least equal, level of performance than that of the control group. This shows that witness ratings always help produce more precise trust evaluation, even in continually changing environments.

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 shows that, by employing the CR component, the CR group outperforms the control group by at least 1.0 utility units in all interactions in the first three experiments (dynamism levels 0, 1, and 2) and in most interactions in the last experiment (dynamism level 3). As can be seen, the dynamism of the environments still affects the performance of the CR group as in the previous set of experiments (the CR group also employs the IT and WR components). However, the CR group proves the efficiency of the CR component by obtaining higher utility than the control group in all the experiments. Of particular important is that the CR group achieves its high level of performance right from the very first interactions. This improvement shows that CR information from the providers does indeed help FIRE to produce a more precise trust measure right from the start of an agent's life (whereas the IT and WR components can perform inefficiently due to the scarcity of trust information).

In summary, we can see that taking various sources of trust information into account helped FIRE produce trust evaluation in a wide variety of situations and that all components contribute to its overall performance.

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 the scale of -1, 0, or $+1$, which means positive, neutral and negative rating respectively. 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 [13] 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 of limited 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 CR component in FIRE has similarities to trust policy management engines such as PolicyMaker [5] and Trust-Serv [11]. These engines grant rights to an agent based on its certificates of its identity according to predefined policies (i.e. rules, such as 'if a is a registered user and it possesses a valid credit card then it can book flights). In contrast, our CR component computationally evaluates (rating) information provided by an agent to deduce its trustworthiness for selecting interaction partners. Certified ratings in FIRE's CR component are also similar to the concept of endorsements in [8] – certificates endorsing that a service (provider) is trusted and preferred by their issuers. Obviously, certified ratings provide the possibility of a richer multitude of rating values. Moreover, in our model, the ratings are aggregated into a CR value, which is then integrated into FIRE's overall trust value. This is not done in [8] because in that work endorsements are viewed as separate from a service's reputation.

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 in a simulated open MAS. By vary-

ing the level of dynamism of the testbed, we also showed that FIRE is able to quickly adapt to a changing environment while still maintaining a high level of performance.

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. In terms of improving its overall performance, we plan to incorporate learning abilities. At present, FIRE is a static parametric model (i.e. all its parameters are set by users in order to suit a particular application domain). This is clearly limiting and so we aim to study which of FIRE's parameters can be adjusted dynamically to adapt it to changes in an agent environment. For instance, if the number of lying agents in its environment increases, an agent may reduce the component coefficient of witness and certified reputation (W_W and W_C, see Section 2.5); or if its environment changes too quickly (e.g. the agents alter their behaviours frequently), it can reduce the local rating history size H to discard older (and less relevant) ratings.

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 D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Department of Sociology, University of Oxford, electronic edition, 2000.

[5] T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4), 2000.

[6] D. Huynh, N. R. Jennings, and N. R. Shadbolt. FIRE: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, 2004.

[7] N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, April 2001.

[8] E. M. Maximilien and M. P. Singh. Reputation and endorsement for web services. *ACM SIGEcom Exchanges*, 3(1):24–31, 2002. ACM Special Interest Group on E-Commerce.

[9] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 2004.

[10] J. Sabater. *Trust and Reputation for Agent Societies*. Phd thesis, Universitat Autnoma de Barcelona, 2003.

[11] H. Skogsrud, B. Benatallah, and F. Casati. Model-driven trust negotiation for web services. *IEEE Internet Computing*, 7(6):45–52, 2003.

[12] 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.

[13] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–908, 2000.