Incorporating Mitigating Circumstances into Reputation Assessment

Simon Miles[1] and Nathan Griffiths[2]

[1] King's College London, London, UK
simon.miles@kcl.ac.uk
[2] University of Warwick, Coventry, UK
nathan.griffiths@warwick.ac.uk

Abstract. Reputation enables customers to select between providers, and balance risk against other aspects of service provision. For new providers that have yet to establish a track record, negative ratings can significantly impact on their chances of being selected. Existing work has shown that malicious or inaccurate reviews, and subjective differences, can be accounted for. However, an honest balanced review of service provision may still be an unreliable predictor of future performance if the circumstances differ. Specifically, mitigating circumstances may have affected previous provision. For example, while a delivery service may generally be reliable, a particular delivery may be delayed by unexpected flooding. A common way to ameliorate such effects is by weighting the influence of past events on reputation by their recency. In this paper, we argue that it is more effective to query detailed records of service provision, using patterns that describe the circumstances to determine the significance of previous interactions.

Keywords: Reputation, Trust, Provenance, Circumstances

1 Introduction

In online service-oriented systems, an accurate assessment of reputation is essential for selecting between alternative providers. Existing methods for reputation assessment have focused on coping with malicious or inaccurate ratings, and with subjective differences, and do not consider the full interaction history and context. The context of previous interactions contains information that could be valuable for reputation assessment. For example, there may have been mitigating circumstances for past failures, such as where a freak event affected provision, or a previously unreliable sub-provider has been replaced. Existing methods do not fully take into account the circumstances in which agents have previously acted, meaning that assessments may not reflect the current circumstances, and so be poor predictors of future interactions. In this paper, we present a reputation assessment method based on querying detailed records of service provision, using patterns that describe the circumstances to determine the relevance of past interactions. Employing a standard provenance model for describing these circumstances, gives a practical means for agents to model, record and query the past. Specifically, the contributions of this paper are as follows.

- A provenance-based approach, with accompanying architecture, to reputation assessment informed by rich information on past service provision.
- Query pattern definitions that characterise common mitigating circumstances and other distinguishing past situations relevant to reputation assessment.
- An extension of an existing reputation assessment algorithm (FIRE [7]) that takes account of the richer information provided in our approach.
- An evaluation of our approach compared to FIRE.

An overview of our approach, with an example circumstance pattern and a high-level evaluation, appears in [10]. This paper extends that work, presenting an in-depth description of the approach and architecture for provenance-based reputation, additional circumstance patterns, and more extensive evaluation.

Reputation and trust are closely related concepts, and there is a lack of consensus in the community regarding the distinction between them [11]. For clarity, in this paper we use the term *reputation* to encompass the concepts variously referred to as trust and reputation in the literature.

We discuss related work in the following section, before presenting our approach in Section 3. The baseline reputation model is described in Section 4 and we present example circumstance patterns in Section 5. Evaluation results are described in Section 6 and our conclusions in Section 7.

2 Background

Given the importance of reputation in real-world environments, there continues to be active research interest in the area. There are several effective computational reputation models, such as ReGreT [13], FIRE [7], TRAVOS [16] and HABIT [15] that draw on direct and indirect experiences to obtain numerical or probabilistic representations for reputation. In dynamic environments, where social relationships evolve and the population changes, it can be difficult to assess reputation as there may be a lack of evidence [1, 7, 8, 14]. Stereotypes provide a useful bootstrapping mechanism, but there needs to be a sufficient evidence base from which to induce a prediction model [1, 3, 14, 18]

Where there is little data for assessing reputation, individual pieces of evidence can carry great weight and, where negative, may cause a provider rarely to be selected, and never be given the opportunity to build their reputation. While reviewer honesty can be tested from past behaviour and dishonest reviews ignored, it is possible for a review to be accurately negative, because of poor service provision, and still not be an accurate predictor of future behaviour. These are examples of *mitigating circumstances*, where the *context* of service provision rather than an agent's ability meant that it was poorly provided, but that context was temporary. Many approaches use *recency* to ameliorate such effects. However, we argue that recency is a blunt instrument. First, recent provision

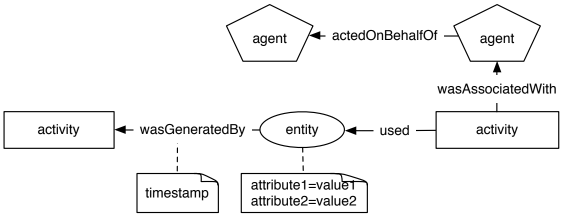


Fig. 1. PROV graph illustrating the key elements

may have been affected by mitigating circumstances, and recency will weight the results higher than older but more accurate data. Second, older interactions may remain good predictors of reliability, because of comparable circumstances.

Instead, we argue for the circumstances of past interactions to be recorded and taken into account more explicitly. This raises the question of what form these records should take, and who should record them. In order to share interaction records between agents, they must be recorded in a commonly interpretable format. PROV is a W3C standard for modelling, serialising and accessing *provenance information*, the history of processes [19]. A PROV document describes in a queryable form the causes and effects within a particular past process of a system (such as agents interacting), as a directed graph with annotations. A visualisation of such a graph, showing PROV's key elements, is shown in Fig. 1. In summary, an *activity* is something that has taken place, making *use* of or *generating entities*, which could be data, physical or other things. *Agents* are parties that were responsible for (*associated with*) activities taking place, and one agent may have been *acting on behalf of* another in this responsibility. Activities, entities and agents (graph nodes) may be annotated with key-value *attributes* describing features that the elements had. *Timestamps* can also be added to show when entities were used or generated by activities.

There has been relatively little use of provenance records for reputation. One of the earliest approaches traversed a decision tree with respect to provenance records to measure reputation [12]. Within the domain of information provision, a richer assessment can be obtained by considering the provenance path of information, the trustworthiness of the information itself, and the reliability of the provider to assess reputation [5, 21]. A risk model can be defined that considers the main risk classes and relationships, which can facilitate a detailed risk assessment for an interaction by evaluating the complete provenance path [17].

3 Approach

To enable the use of provenance records to provide personalised reputation assessments, we have proposed the architecture illustrated in Fig. 2, in which clients make requests to an *assessor* for reputation assessments [6]. The assessor relies on provenance graphs to determine reputation, rather than on individual or third party ratings as in existing work. Provenance records are recorded as a side-effect

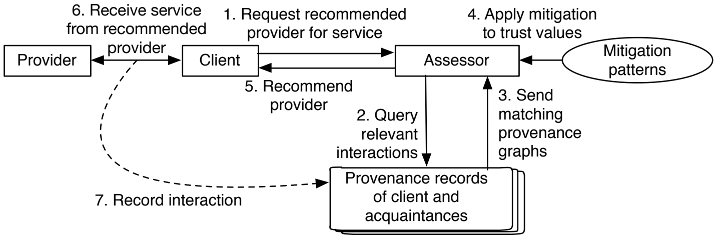

6. Receive service from recommended provider
1. Request recommended provider for service
4. Apply mitigation to trust values
Provider
Client
Assessor
Mitigation patterns
5. Recommend provider
2. Query relevant interactions
3. Send matching provenance graphs
Provenance records of client and acquaintances
7. Record interaction

Fig. 2. An architecture for provenance-based service provider reputation

of interactions, by one or multiple parties, providing crucial evidence that may be missing for assessing reputation [2]. For example, in a logistics chain in addition to clients recording information, providers can record information about sub-contractors, giving information about sub-contractors' performance.

This allows mitigation, situation, indirect responsibility, and other such context to be accounted for, and the interdependencies of providers to be understood. Mitigation can have many forms, such as a subsequently replaced subcontractor failing to deliver on time, or a client failing to specify required conditions (e.g. expiration date of goods being shipped). The assessor looks for patterns in the provenance that indicate situations relevant to the current client's needs and mitigating circumstances affecting the providers. Provenance data is suitable for this because it includes the causal connections between interactions, and so captures the dependencies between agents' actions. It can include multiple parties to an interaction and their organisational connections. The assessor filters the provenance for key subgraphs from which reputation can be assessed using existing approaches, by identifying successful and failed interactions and adjusting these by mitigation and situation relevance. Assessing reputation in this way avoids the problem of when to update trust, as whenever an assessment is required it is determined using all available evidence.

Reputation enables the assessment and management of the risk associated with interacting with others, and enables agents to balance risk against factors such as cost when considering alternative providers. Such environments can be viewed as service-oriented systems, in which agents provide and consume services. We take an abstract view of service-oriented systems, without prescribing a particular technology. We assume that there are mechanisms for service advertisement and for service discovery. We also assume that service adverts can optionally include details of provision, such as specifiying particular sub-providers if appropriate. Finally, we assume that agents record details of their interactions in the form of provenance records, which can be used to assess reputation. The practicality of this last requirement is discussed in Section 6.3.

4 Baseline Reputation

Provenance records not only contain rich information that enable reasoning about aspects such as mitigating circumstances, but they also provide a means to maximise the amount of information available for reputation assessment. In this section, we describe how reputation can be driven by provenance records. For the purposes of illustration we consider FIRE [7], but note that other approaches, such as those discussed in Section 2 or machine learning techniques, can similarly be adapted to use provenance records.

4.1 The FIRE Reputation Model

FIRE combines four different types of reputation and trust: interaction trust from direct experience, witness reputation from third party reports, role-based trust, and certified reputation based on third-party references [7]. The direct experience and witness reputation components are based on ReGreT [13]. In this paper our focus is on using provenance records of interactions to support reputation, and on defining query patterns for mitigating circumstances. Role-based trust and certified reputation are tangential to this focus, as they are not directly based on interaction records. Therefore, we do not consider role-based trust and certified reputation in this paper (although we do not argue against their usefulness). Reputation is assessed in FIRE from *rating* tuples of the form (a, b, c, i, v), where a and b are agents that participated in interaction i such that a gave b a rating of $v \in [-1, +1]$ for the term c (e.g. reliability, quality, timeliness). A rating of $+1$ is absolutely positive, -1 is absolutely negative, and 0 is neutral. In FIRE, each agent has a history size H and stores the last H ratings it has given in its local database. FIRE gives more weight to recent interactions using a *rating weight function*, ω_K, for each type of reputation, where $K \in \{I, W\}$ representing interaction trust and witness reputation respectively.

The trust value agent a has in b with respect to term c is calculated as the weighted mean of the available ratings:

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

where $\mathcal{R}_K(a, b, c)$ is the set of ratings stored by a regarding b for component K, and v_i is the value of rating r_i.

To determine direct interaction reputation an assessing agent a extracts the set of ratings, $\mathcal{R}_K(a, b, c)$, from its database that have the form $(a, b, c, \_, \_)$ where b is the agent being assessed, c is the term of interest, and "_" matches any value. These ratings are scaled using a *rating recency factor*, λ, in the rating weight function, and combined using Equation 1. FIRE instantiates the rating weight function for interaction trust as:

$$\omega_I(r_i) = e^{\frac{\Delta t(r_i)}{\lambda}} \tag{2}$$

where $\omega_I(r_i)$ is the weight for rating r_i and $\Delta t(r_i)$ the time since r_i was recorded.

Agents maintain a list of acquaintances, and use these to identify witnesses in order to evaluate witness reputation. Specifically, an evaluator a will ask its acquaintances for ratings of b for term c, who either return a rating or pass on the request to their acquaintances if they have not interacted with b. FIRE uses a variation of Yu and Singh's referral system [22], with parameters to limit the branching factor and referral length to limit the propagation of requests. The ratings obtained from referrals are then used to calculate witness reputation (using Equation 1, with $\omega_W(r_i) = \omega_I(r_i)$). FIRE assumes that agents are willing to help find witness ratings, and that ratings are honest and credible. In general, these assumptions may not hold and $\omega_W(r_i)$ should account for credibility.

The overall term trust in an agent is calculated as a weighted mean of the component sources:

$$\mathcal{T}(a,b,c) = \frac{\sum_{K \in \{I,W\}} \omega_K \cdot \mathcal{T}_K(a,b,c)}{\sum_{K \in \{I,W\}} \omega_K} \tag{3}$$

where the reliability of the reputation value for component K is $\rho_K(a,b,c)$, $\omega_K = W_K \cdot \rho_K(a,b,c)$, and W_I and W_W are parameters that determine the importance of each component. The reliability of a reputation value is determined by a combination of the rating reliability and deviation reliability, which characterise a reputation assessment in terms of the number and variability of the ratings on which it is based. The calculations are beyond the scope of this paper (details can be found in [7]), but we note that these metrics can also be calculated from the information in the provenance records.

FIRE does not specify how reputation for different terms is combined into an overall assessment. For simplicity, we assume that terms have equal weight in the same normalised units, and we average across ratings for all terms relevant to a service. Applying varying weights would be a trivial extension.

4.2 Reputation from Provenance Records

As provenance records are not simple tuples containing ratings, unlike in FIRE, we need to determine whether an interaction was good or bad. An interaction's quality could be measured in different terms: the adequacy of the product or service, the speed with which the service was provided, etc. Different terms correspond to different features of provenance graphs. For example, PROV allows timestamps to be added to *use* relations (when an entity began being used by an activity), generation relations (when an entity was generated by an activity), and the start and end of activities. Two timestamps of interest in service provision are the use of the client's request by the service provider, i.e. when the service was requested, and the generation of the service result by the provider, i.e. when the service was completed. Subtracting one from the other gives the duration of service provision. Comparison of this period to the client's expectation gives a rating for the interaction's timeliness term.

Another term could be an observable quality of a product, for example whether a product is damaged. By querying the relevant attribute of the product of a service, a rating can be determined for the quality term. A more interesting term could be the proportion of the product made from materials from sustainable sources. Determining a rating for this latter property would require looking across multiple parts of the provenance graph for an interaction, to determine the sustainability of each component part of the eventual product. For example, to determine the sustainability of a garment details of the fabric and raw materials (e.g. cotton, dye, and fasteners) must also be evaluated. Terms are often domain-specific and are not further discussed here.

5 Circumstance Patterns

PROV data describes past processes as causal graphs, captured from multiple parties and interlinked. The interactions which comprise a service being provided can be described by a sub-graph, and inspecting features of the sub-graphs, such as through a SPARQL query [20], can determine the extent to which they inform reputation. In this section, we specify three mitigating circumstances patterns that could be detected in provenance data. These examples are not intended to be exhaustive, but illustrate the form of such patterns in our approach.

5.1 Unreliable Sub-provider

In the first mitigating circumstance, a provider's poor service on a past occasion was due to reliance on a poor sub-provider for some aspect of the service. If the provider has changed sub-provider, the past interaction should not be considered relevant to their current reputation.[3] This is a richer way of accounting for sub-provider actions than simply discounting based on position in a delegation chain [4]. In other words, Provider A's reputation should account for the fact that previous poor service was due to Provider A relying on Provider B, who they no longer use. The provenance should show:

1. Provider B was used where there was poor service provision,
2. Provider B's activities were the likely cause of the poor provision, and
3. Provider A no longer uses Provider B (not necessarily shown through provenance).

A provenance pattern showing reliance on a sub-provider in a particular instance can be defined as follows. For reference, activities are labelled with An (where n is a number) and entities are labelled with En. Fig. 3 illustrates this pattern, along with some of the specific cases below.

Step 1 A client process, A1, sends a request, E1, for a service to a process, A2, for which Provider A is responsible. In the PROV graph, this means that E1 wasGeneratedBy A1, A2 used E1, and A2 wasAssociatedWith Provider A.

Step 2 A2 sends a request, E2, to a service process, A3, for which Provider B is responsible. In the PROV graph, this means that E2 wasGeneratedBy A2, A3 used E2, and A3 wasAssociatedWith Provider B.

[3] Such a situation may indicate poor judgement and so have a degree of relevance, but this is not considered in this paper.

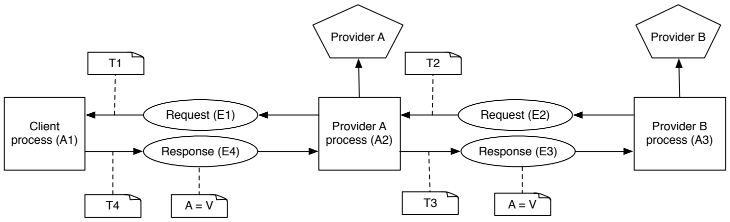


Fig. 3. Provenance graph pattern for unreliable sub-provider circumstance

Step 3 A3 completes the action and sends a result, E3, back to A2. In the PROV graph, this means that E3 wasGeneratedBy A3, and A2 used E3.

Step 4 A2 completes the service provision, sending the result, E4, back to A1, so that the client has received the service requested. In the PROV graph, this means that E4 wasGeneratedBy A2, and A1 used E4.

We can distinguish cases in which Provider B would be the likely cause of poor quality service provision. Each case corresponds to an extension of the above provenance pattern.

Case 1. An aspect of the result of provision is poor, and that aspect is apparent in Provider B's contribution. For example, Provider A may have provided a website for a company which appears poor due to low resolution images supplied by Provider B. The extensions to the original pattern are as follows.

- The service provision result, E4, has an attribute A=V, which is a reason for the result being poor (e.g. resolution=low).
- The intermediate result from Provider B, E3, has this same attribute A=V.

Case 2. The poor provision may not be due to eventual outcome but due to the time taken to provide the service, and this can be shown to be due to the slowness of Provider B. The extensions to the original pattern are as follows.

- The sending of the service request (i.e. the relation E1 wasGeneratedBy A1), is timestamped with T1.
- The receipt of the service result (i.e. the relation A1 used E4), is timestamped with T4.
- The sending of the delegated request (i.e. the relation E2 wasGeneratedBy A2), is timestamped with T2.
- The receipt of the delegated service result (i.e. the relation A2 used E3), is timestamped with T3.
- $T4 - T1 > X$, where X is the reasonable upper limit for the service to be provided, and $T3 - T2 > Y$, where Y is some significant portion of X.

The final criterion required for the above patterns to affect Provider A's reputation assessment is to show that Provider A no longer uses Provider B. This could be through (i) recent provenance of Provider A's provision showing no use of Provider B, or (ii) Provider A's advert for their service specifying which sub-provider they currently use. The latter is assumed the in evaluation below.

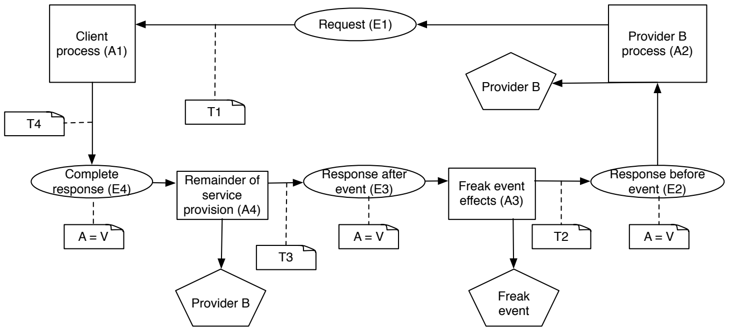


Fig. 4. Provenance graph pattern for freak event circumstance

We also note that a variation of this pattern is also useful, namely to identify situations in which successful service provision was due to a *good* sub-provider who is no longer used. In this variation the same pattern is used but with poor provision replaced by good provision.

5.2 Freak Event

In the second circumstance, the service provision of Provider A was affected by a one-off substantial event, e.g. ash from an erupting volcano, flooding blocking roads, etc. The freak event can be considered to be an agent in the provenance graph, as it is an autonomously acting entity. The provenance should show:

1. The effects of a known freak event were part of the process of Provider A providing the service, and
2. The part of the process affected by the freak event was the likely cause of the poor service.

The pattern should show that the effects of the freak event were part of the service provision process, illustrated in Fig. 4.

Step 1 A client process, A1, sends a request, E1, for a service to A2 for which Provider B is responsible. In the provenance graph, this means that E1 wasGeneratedBy A1, A2 used E1, and A2 wasAssociatedWith Provider B.

Step 2 A2 begins providing the service by producing entity E2. E2 wasGeneratedBy A2.

Step 3 The relevant effects, A3, of the freak event affect the service provision, so we distinguish what is provided before those effects, E2, and after, E3. A3 used E2, E3 wasGeneratedBy A3, A3 wasAssociatedWith the freak event.

Step 4 The remainder of the service provision process, A4, completes from the state after the freak event has affected the process, E3, and produces the final provision result, E4. A4 used E3, E4 wasGeneratedBy A4.

Step 5 Finally, provision is completed and returned to the client. A1 used E4.

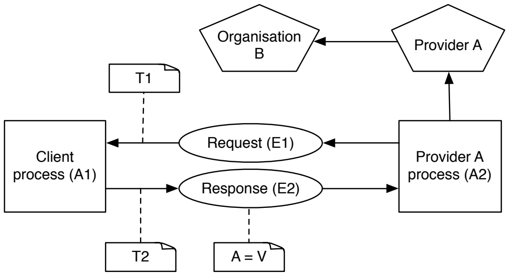


Fig. 5. Provenance graph pattern for poor organisation culture circumstance

Similar to the first circumstance above, we can distinguish the cases in which the freak event is the likely cause of eventual poor service. The attributes can indicate that the product before the event (E2) was high quality, while after it (E3) was not, e.g. water damage affecting a parcel. Any delay between the request and response could be primarily due to the freak event (A3).

5.3 Poor Organisation Culture

In the third case, Provider A may be an individual within Organisation B. In such cases, the culture of the organisation affects the individual and the effectiveness of the individual affects the organisation. If Provider A leaves the organisation, this past relationship should be taken into account: Provider A may operate differently in a different organisational culture. The provenance should show:

1. Provider A provided poor service while working for Organisation B, and
2. Provider A is no longer working for Organisation B.

A provenance pattern showing provision of a service within an organisation in a particular instance could be as follows (illustrated in Fig. 5).

Step 1 A client process, A1, sends a request, E1, for a service to A2, for which Provider A is responsible. In the provenance graph, this means that E1 wasGeneratedBy A1, A2 used E1, and A2 wasAssociatedWith Provider A.

Step 2 Provider A is acting on behalf of Organisation B in performing A2. In the provenance graph, this means Provider A actedOnBehalfOf Organisation B in its responsibility for A2 (the latter not depicted Fig. 5 to retain clarity).

Step 3 A2 completes the service provision sending the result, E2, back to A1, so that the client has received the service requested. In the provenance graph, this means that E2 wasGeneratedBy A2, and A1 used E2.

We can then distinguish the cases in which the culture of Organisation B may be a mitigating factor in Provider A's poor provision. Poor performance is identified as described above: either an attribute indicating low quality, a part that is of low quality, or too long a period between the request and response. A variation on the circumstance is to observe where agents were, but are no longer, employed by organisations with a *good* culture.

6 Evaluation

We evaluated our approach through simulation, comparing it with FIRE, using an environment based on that used in the original evaluation of FIRE [7]. For transparency, the simulation code is published as open source[4].

6.1 Extending FIRE

Existing reputation methods do not account for mitigating circumstances and the context of service provision. The *context* of an interaction is not considered and there is no mechanism for considering mitigating circumstances. In our approach, each agent has its own provenance store, and to determine the reputation of a provider on behalf of a client the assessor queries that client's provenance store and those of its acquaintances. For each interaction recorded in the provenance stores the outcome is considered according to the term(s) that the client is interested in. Since, for illustration, we adopt the FIRE model, the assessor extracts ratings from the provenance of the form $(\_, b, c, i, v)$, where b is the provider in interaction i, and the client in i gave b a rating of v for term c. These ratings are then used to determine reputation (using Equations 1 and 3).

Mitigating circumstances and context can be incorporated into existing reputation models by adjusting the weighting that is given to the rating resulting from an interaction for which there are mitigating circumstances. In FIRE, this can be done through the rating weight function, ω_K, for each type of reputation, where $K \in \{I, W\}$, by a factor that accounts for mitigation, specifically:

$$\omega_I(r_i) = \omega_W(r_i) = m \qquad (4)$$

where m is the mitigation weight factor. This factor reflects how convincing an agent considers particular mitigating circumstances, and is defined on a per pattern basis. For the sub-provider and organisation patterns this corresponds to the perceived contribution of a sub-provider or organisation to the service provision, while for a freak event it corresponds to the perceived impact of the event. Mitigation weight factors can be estimated from knowledge of the system and each agent can ascribe a mitigation value to each of its mitigating circumstance patterns. For simplicity, however, we ascribe a global value to each pattern.

Our FIRE implementation calculates trust on the basis of individual and witness experience, i.e. a client's provenance records and those of its acquaintances, applying equal weight to each, but we exclude role-based and certified trust as discussed in Section 4. The original evaluation of FIRE allows *exploration* of the space of providers, meaning that the most trusted provider is not always chosen. We include an exploration probability, e, where a client selects the most trusted provider with probability $1 - e$, else will select the next most trusted with probability $1 - e$, etc. This differs from the original evaluation of FIRE which uses Boltzmann exploration to reduce exploration over time. The

[4] http://bit.ly/1uqLAZO

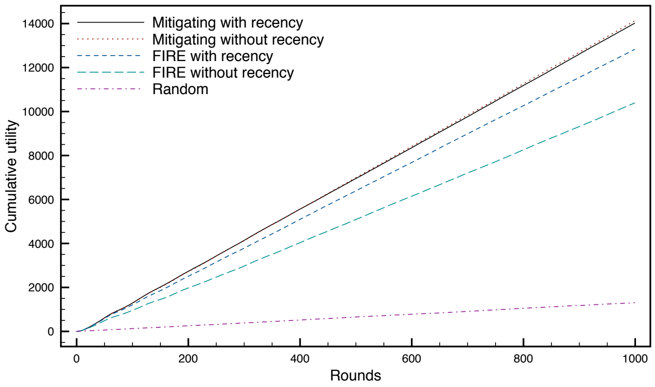


Fig. 6. Cumulative utility over time for all mitigating circumstances patterns.

effectiveness of Boltzmann exploration requires the best action to be well separated from others [9]. This is not a reasonable assumption, since providers may be similarly trustworthy. Moreover, there is an assumption that convergence is possible, and in a dynamic environment this is not appropriate.

FIRE's original evaluation divided agents into clients and providers, whereas we assume any agent can be a client or provider. To improve simulation performance we set a memory limit such that, by FIRE's recency weighting, records with a weighting of $\leq 1\%$ are not retained.

6.2 Results

We evaluated the strategies on a simulated network of 100 agents providing services to each other over 1000 rounds. Agents are positioned on, and explore, a spherical world which dictates their neighbours and acquaintances (as in the original evaluation of FIRE [7]), with an average of around 3 neighbours each. This means agents tended to form 2 to 4 clusters of acquaintances. There were 5 primary capabilities (types of service which may require sub-capabilities), capabilities have two terms (quality and timeliness), and each agent has 3 capabilities. Each agent has a 50% chance to request a service each round and 20% chance not to pick the most trusted agent. Agents switch sub-provider every 1–15 rounds. Freak events occur with 25% probability and affected interactions are weighted at 25% relevancy by our strategy. Where recency scaling was applied, it was set such that after 5 rounds it is 50% weight. There are 10 organisations, 30% with a poor culture, reducing the terms of the services provided, while 70% had a good culture. Agents change organisation every 1–15 rounds. The utility gained in a round is the sum of utility gained per service provision, where the latter is the average of quality and timeliness of the provision (each in $[-1, 1]$).

We compared five strategies: FIRE, our approach (Mitigating) with and without recency, FIRE without recency, and random selection. Each strategy was evaluated in 50 networks and the results averaged. Fig. 6 shows the results where all three example circumstances are present (poor sub-providers, freak

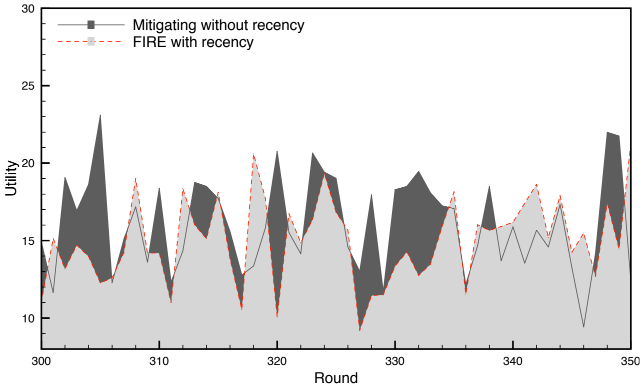
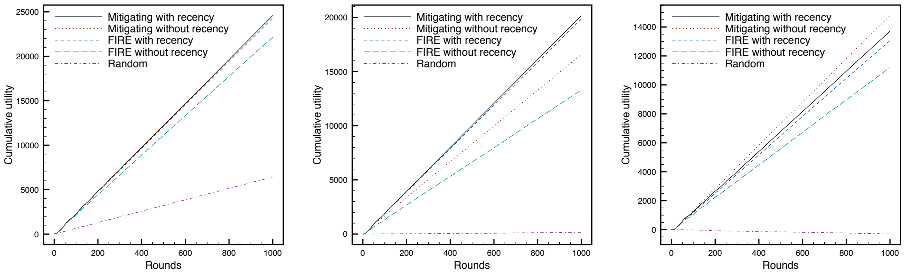


Fig. 7. Per-round utility over one simulation

events, poor organisational culture). Our approach has improved performance, both with and without recency, over FIRE, with an improvement of 10.1% without and 9.3% with recency scaling respectively. The recency scaling of FIRE is also shown to be beneficial where mitigating circumstances are not taken into account, i.e. FIRE is better than FIRE without recency. These results match the intuition that recency is valuable for taking account of changes in circumstances, but is crude compared to what is possible when past circumstances are visible. When recency is combined with mitigating circumstances there is negligible improvement, further supporting this intuition.

We also considered how utility varied over a simulation, to better understand the results above. Fig. 7 shows the per-round utility for an extract of a single simulation for FIRE and our approach without recency (other approaches are omitted for clarity). Utility varies significantly over time, as changing circumstances mean the most trusted agents may not be the best providers. Our approach has more and higher peaks than FIRE, leading to the higher cumulative utility described above. We believe that this is because our strategy recovers from a change in circumstance more quickly than FIRE. While FIRE's recency scaling means that irrelevant past circumstances are eventually ignored, our approach immediately takes account of the difference in past and present circumstances.

To understand how individual circumstances contributed to the results, we simulated the system with a single circumstance pattern applied. In the case of freak events (Fig. 8a) our approach performs similarly to FIRE, with a small improvement (1.1% in cumulative utility over 1000 rounds). As expected, FIRE without recency performs worse. Our approach has similar results with and without recency, implying that for a low incidence of freak events (25%), consideration of recency along with mitigating circumstances has little effect. For unreliable sub-providers (Fig. 8b), there is value to scaling by recency in addition to considering mitigating circumstances. Our approach with recency performs similarly to FIRE (with a 1.6% improvement), but without recency scaling the utility is significantly lower. Note that both variants of the sub-provider pattern are used, and both poor and good interactions are scaled. With poor organisation



(a) Freak event (b) Unreliable sub-provider (c) Poor organisation culture

Fig. 8. Cumulative utility for use of the individual mitigating circumstances patterns.

culture (Fig. 8c) our approach, with and without recency, outperforms FIRE, with the largest improvement without recency (13.2%). Here recency scaling reduces performance, and we believe this is because the pattern identifies appropriate situations, and additional scaling reduces the impact of relevant ratings.

6.3 Discussion

In this section, we attempt to answer questions about the results and approach.

Why does accounting for recency seem to be a disadvantage in some results? Recency accounts for changes between the past and present, allowing obsolete information to be forgotten. Weighting relevance by matching against the current circumstance based on provenance patterns aims to account for the past more precisely. Therefore, where the circumstance patterns work as expected, also accounting for recency will dilute the precision, producing worse results.

Why does the result with just unreliable sub-providers show a disadvantage for our approach? The results in Figure 8b show our strategy without recency being outperformed by our strategy with recency and FIRE. As discussed above, this suggests that the current pattern used for this circumstance does not provide the correct relevance weighting to account for the past precisely, and so recency is a valuable approximation. We have not yet determined why this pattern is imprecise, and it is under investigation.

Why would providers capture provenance graphs? In a practical system, we must account for why provenance graphs would be captured and how they would be accessed by clients. Providers are the obvious source of the provenance data, as it is a record of service provision, but it may be against their interests to release records of poor performance. There are a few answers to this question, though full exploration of the issue is beyond the scope of this paper. First, contractual agreements between clients and providers can require some recording of details as part of providing the service, possibly with involvement of a notary to help ensure validity. In many domains such documentation is a contractual obligation, e.g. journalists must document evidence capture and financial services must document processes for audit. Second, the entities in the provenance graphs are generally exchanged in messages between parties, so there are two agents that can verify the entities were as documented (a commonly used mechanism for non-repudiation). Finally, at a minimum, some information should be present in the client-accessible service advert at the time of service provision, e.g. the organisation to which the provider belongs or sub-provider they use.

What is the value of using PROV graphs over simpler forms? The information recorded in each circumstance (sub-provider, organisation, freak event, etc.) could be provided in a simpler form than a PROV graph, e.g. a tuple. However, a PROV graph is of more practical value. First, every circumstance is different and there may be a varied set of circumstances considered over time, so a single typed tuple is inadequate. Second, the contents of provenance graphs can be collated from data recorded by a set of independent agents, and so it is essential that the provenance follows a standard (W3C PROV). Third, and related, by using PROV there are defined serialisations which mean that clients have a standard means to query the data, e.g. by SPARQL over RDF PROV.

7 Conclusions

In this paper we have described how provenance records can be used to provide the information needed to assess reputation. We have shown how provenance records can be queried to identify when mitigating circumstances occur, to account for context, and argue that this is a more principled approach than simply scaling by recency. Specifically, we defined query patterns for unreliable sub-providers, freak events, and poor organisational culture. The approach is agnostic regarding the reputation model, but for the purposes of evaluation we adopted FIRE [7]. Our evaluation shows that consideration of mitigating circumstances improves performance, but that it is crucial for query patterns to fully capture the context otherwise recency scaling is still required. Future work will define additional query patterns, and develop a method for providing rationale from provenance records explaining reputation assessment.

Acknowledgements This work was part funded by the UK Engineering and Physical Sciences Research Council as part of the Justified Assessments of Service Provider Reputation project, ref. EP/M012654/1 and EP/M012662/1.

References

1. C. Burnett, T. J. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *Proc. of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 241–248, 2010.
2. C. Burnett, T. J. Norman, and K. Sycara. Trust decision-making in multi-agent systems. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence*, pages 115–120, 2011.
3. C. Burnett, T. J. Norman, K. Sycara, and N. Oren. Supporting trust assessment and decision-making in coalitions. *IEEE Intelligent Systems*, in press.

4. C. Burnett and N. Oren. Sub-delegation and trust. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 1359–1360, 2012.
5. C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu. An approach to evaluate data trustworthiness based on data provenance. In *Proc. of the 5th VLDB workshop on Secure Data Management*, pages 82–98, 2008.
6. N. Griffiths and S. Miles. An architecture for justified assessments of service provider reputation. In *Proc. of the 10th IEEE Int. Conf. on e-Business Engineering*, pages 345–352, 2013.
7. T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *J. of Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
8. A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43:618–644, 2007.
9. L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *J. of Artificial Intelligence Research*, 4:237–285, 1996.
10. S. Miles and N. Griffiths. Accounting for circumstances in reputation assessment. In *Proc of the 14th Int. Conf. on Autonomous Agents and Multiagent Systems*, 2015.
11. I. Pinyol and J. Sabater-Mir. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, 40:1–25, 2013.
12. S. Rajbhandari, A. Contes, O. F. Rana, et al. Trust assessment using provenance in service oriented applications. In *Proc. of the 10th IEEE Int. Enterprise Distributed Object Computing Conference Workshops*, page 65, 2006.
13. J. Sabater. Evaluating the ReGreT system. *Applied Artificial Intelligence*, 18(9-10):797–813, 2004.
14. M. Sensoy, B. Yilmaz, and T. J. Norman. STAGE: Stereotypical Trust Assessment Through Graph Extraction. *Computational Intelligence*, 2014.
15. W. T. L. Teacy, M. Luck, A. Rogers, and N. R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*, 193:149–185, 2012.
16. W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *Proc. of the 4th Int. Conf. on Autonomous Agents and Multiagent Systems*, pages 997–1004, 2005.
17. P. Townend, D. Webster, C. C. Venters, et al. Personalised provenance reasoning models and risk assessment in business systems: A case study. In *Proc. of the 7th IEEE Int. Symposium on Service Oriented System Engineering*, pages 329–334, 2013.
18. J. Urbano, A. P. Roacha, and E. Oliveira. Refining the trustworthiness assessment of suppliers through extraction of stereotypes. In *Proc. of the 12th Int. Conf. on Enterprise Information Systems*, pages 85–92, 2010.
19. W3C. PROV model primer. http://www.w3.org/TR/prov-primer/, 2013.
20. W3C. Sparql 1.1 overview. http://www.w3.org/TR/sparql11-overview/, 2013.
21. X. Wang, K. Govindan, and P. Mohapatra. Provenance-based information trustworthiness evaluation in multi-hop networks. In *Proc. of the IEEE Global Telecommunications Conference*, pages 1–5, 2010.
22. B. Yu and M. P. Singh. Searching social networks. In *Proc. of the 2nd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems*, pages 65–72, 2003.