

Context-driven Assessment of Provider Reputation in Composite Provision Scenarios

Lina Barakat¹, Phillip Taylor², Nathan Griffiths², and Simon Miles¹

¹ King's College London, United Kingdom
{lina.barakat,simon.miles}@kcl.ac.uk

² University of Warwick, United Kingdom
{Phillip.Taylor,Nathan.Griffiths}@warwick.ac.uk

Abstract. Service-oriented computing has become the de-facto way of developing distributed applications and, in such systems, an accurate assessment of reputation is essential for selecting between alternative providers. Existing methods typically assess reputation on a combination of direct experiences by the client being provided with a service and third party recommendations, but they exclude from consideration a wealth of information about the context of providers' previous actions. Such information is particularly important in composite service provision scenarios, where providers may *delegate* sub-tasks to others, and thus their success or failure needs to be interpreted in this context and reputation assessed according to responsibility. In response, to enable richer, more accurate reputation mechanisms, this paper models the delegation knowledge underlying a composite service provision, and incorporates such knowledge into the reputation assessment process, adjusting the contributions of past interactions with the composite service provider according to delegation context relevance. Experimental results demonstrate the effectiveness of the proposed approach.

Keywords: reputation assessment, delegation context, composite service provider, interaction weighting

1 Introduction

A service-oriented system can be seen as a dynamic marketplace, where individuals and organisations rely on providers to execute services with an appropriate quality in order to fulfil their own goals. Such reliance implies a degree of risk through dependence upon a third party, and so there is a need for mechanisms that ensure correctness and fairness of providers' and customers' behaviour to help assess and manage this risk. Trust and reputation are concepts commonly modelled in mechanisms for improving the success of interactions by minimising uncertainty when self-interested individuals interact [11]. Trust is an assessment of the likelihood that an individual or organisation will cooperate and fulfil its commitments [12], while reputation can be viewed as the public perception of the trustworthiness of a given entity [13]. In this paper, we will use the terms trust and reputation interchangeably.

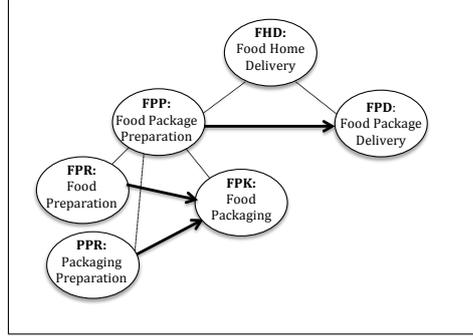


Fig. 1. Delegation Hierarchy of Food Home Delivery

Many models exist in which reputation is derived from direct experience of clients and third party recommendations, with numerical or probabilistic representations for reputation [1–7]. However, these existing methods focus solely on the clients’ experience of the ultimate outcome of a service provision, without consideration of the full history of activity behind such provision, thus omitting from assessment potentially relevant information. Such information is of particular importance in the case of composite service provision, where the provider may depend on other sub-providers to accomplish the task requested.

To illustrate, consider a provider P_{FHD} offering home delivery of nutritious food packages (e.g. meal packages) to customers. The provider does not produce the food packages locally, but deals with a number of specialised suppliers for this purpose (e.g. professional individuals, food companies, etc). Specifically, provider P_{FHD} exposes the options available from these suppliers to potential customers via a dedicated search interface, collects a customer’s order, and delegates this order to a suitable food package supplier. Following the preparation of the food package, provider P_{FHD} contacts its food transportation partner, provider P_{FPD} , for the delivery of the food package to the customer. Figure 1 depicts the delegation hierarchy of provider P_{FHD} and its sub-providers. This hierarchy is not visible to the customer, who rates its interaction with provider P_{FHD} based on the customer’s perception of the final outcome (the food package delivered). Computing the reputation of provider P_{FHD} , without interrogating the history of delegation underlying customer ratings, might yield an inaccurate reputation assessment for the current situation. For example, a low reputation of provider P_{FHD} due to the past failures of its transportation sub-provider P_{FPD} to meet consumers’ constraints on delivery time might give an unfair view if the sub-provider has been changed to avoid such failures re-occurring.

In response, this paper extends existing reputation models by incorporating the delegation context underlying a composite service provision into the reputation assessment process³. In particular, the delegation context of a provision is utilised as additional evidence for determining the relevance of the ratings

³ See <http://jaspr.org/case-studies> for example usage scenarios

Table 1. Symbols often used throughout the paper

Symbol	Description	Symbol	Description
<i>attr</i>	Capability's attributes	<i>P</i>	Set of providers
<i>C</i>	Set of capabilities	<i>rating</i>	Interaction's rating for an aspect
<i>cmp</i>	Goal (composite) capability	<i>rel</i>	Delegation context relevance
<i>cmd</i>	Capability's candidate providers	<i>rep</i>	Reputation score for an aspect
<i>dctx</i>	Interaction's delegation context	<i>time</i>	Interaction's time
<i>I</i>	Set of interactions	<i>wt</i>	Interaction's weight

available on this provision for the current situation, thus enabling more accurate and informed reputation estimates. The rest of the paper is organised as follows. Section 2 and Section 3 provide the delegation model and the rating model, respectively, for a composite service provider. The incorporation of delegation context into reputation assessment is detailed in Section 4, while a corresponding evaluation is presented in Section 5. Section 6 discusses related issues, Section 7 presents related work, and finally Section 8 concludes the paper.

2 Delegation Model

To achieve a composite task *cmp*, the corresponding provider, henceforth referred to as P_{cmp} , may often rely on a number of sub-contractors to perform the various sub-tasks involved. Formally, knowledge of such a delegation model is a tuple, $(C, P, cmd, dg, attr)$, detailed below (see Table 1 for the often used notation).

C is the set of capabilities relevant for achieving composite task *cmp* (including *cmp* itself). Each capability $c \in C$ denotes a competence in achieving a particular sub-task (simple or composite) of composite task *cmp*. In our example, $C = \{FHD, FPP, FPD, FPR, PPR, FPK\}$. Note that, we refer to a capability and its corresponding task interchangeably. We keep the definition of a capability generic to be applicable to a wide range of domains, e.g. it may refer to an operation signature, a resource specification, or an ontology term.

P is the set of available providers in the community offering capabilities from C . Providers encapsulate such offerings within *services*, and expose them through uniform, machine-readable interfaces (or metadata) on a network of customers.

$cmd : C \rightarrow 2^P$, is the candidate provider function, mapping each capability $c \in C$ to the set of providers $cmd(c) \subset P$ offering this capability as a service. Different mechanisms are possible for discovering candidate providers: by consulting a central service repository (e.g. a UDDI⁴ registry) storing service metadata (e.g. SAWSD⁵ descriptions); or by calling for service proposals over the network (e.g. using the contract net protocol [16]). We make no assumptions in our model about any specific technology or service discovery and matching mechanism.

$dg : C \rightarrow G$, is the decomposition graph function, defining the hierarchical delegation structure in the community. It maps a composite capability $c \in C$ to its decomposition graph $dg(c) = (V, E) \in G$, which specifies the comprising

⁴ http://uddi.org/pubs/uddi_v3.htm#_Toc85907967

⁵ <http://www.w3.org/TR/sawsd1/>

(finer-grained) sub-capabilities that can be outsourced to sub-contractors and their dependencies, such that $V \subset C$, $E \subset V \times V$, and G is the set of all directed graphs that can be formed from the capabilities in C . For instance, in our example, $dg(FHD) = (\{FPP, FPD\}, \{(FPP, FPD)\})$.

Finally, $attr : C \rightarrow 2^{AN}$, is the attribute function, mapping each capability $c \in C$ to the set of attributes $attr(c) \subset AN$ characterising this capability (AN is the set of all attribute names). Such features are domain dependent, and can be either *global* (common to all tasks), e.g. cost and duration, or *local* (only specific to particular tasks), e.g. the quality of packaging (which is specific to packaging-related tasks) in the food manufacturing domain. Note that a composite capability inherits all the attributes characterising its sub-capabilities.

3 Rating Model

The rating model of a composite service provider P_{cmp} (the ratee), enriched with delegation context, is a tuple, $(I, time, rater, rating, dctx)$, as detailed below.

I is the set of previous interactions with the ratee up to the current time step, for which ratings are available to the reputation assessor.

$time : I \rightarrow T \cup \{\perp\}$, is an interaction time step function, specifying for an interaction $i \in I$, the point in time $time(i) \in T$ at which the interaction took place (T is the set of all time steps). Note that $time(i) = \perp$ indicates that the time of interaction i is either not utilised for reputation assessment (the case of some reputation models), or unknown (e.g. knowledge of the interaction is acquired from a third party with no time information).

$rater : I \rightarrow U \cup \{\perp\}$, is an interaction rater function, specifying the rating party $rater(i) \in U$ for interaction $i \in I$ (U is the set of all users who interacted with the ratee). The availability and utilisation of the rater knowledge depend on the reputation model adopted: while some models allow for third party ratings, others only account for personal experience. Moreover, in some models, the rater identities are kept anonymous with the ratings being collected and accessible via a dedicated store, in which case $\forall i \in I, rater(i) = \perp$.

$rating : I \times AN \cup \{overall\} \rightarrow RV \cup \{\perp\}$, is an interaction rating function, mapping an interaction $i \in I$, to the rating, $rating(i, a) \in RV$, provided for this interaction on aspect $a \in attr(cmp) \cup \{overall\}$. Here, *overall* denotes an overall perspective on the interaction, and RV is the set of all possible rating values. Note that $rating(i, a) = \perp$ indicates that rating on aspect a is not available for interaction i . The domain of rating values RV depends on the reputation model adopted, e.g. it can be binary, indicating either success or failure, or numeric, indicating the level of satisfaction according to a particular scale.

Finally, $dctx : I \times C \rightarrow P \cup \{\perp\}$, is an interaction's *delegation context* function (which we introduce into the rating model), representing knowledge regarding the participants in the realisation of composite capability cmp during interaction $i \in I$. In particular, it maps a sub-capability node $c \in C$ in a decomposition graph, to the provider $dctx(i, c) \in cnd(c)$, to which this capability was delegated by the ratee (or its sub-contractors) during interaction i . Details of how such

delegation knowledge can be acquired by the reputation assessor are discussed in Section 6. In general, different levels of visibility might be available to the assessor regarding this knowledge. For example, the assessor might be aware of the delegation hierarchy fully, or only partially (e.g. only the direct sub-contractors of the ratee, performing the sub-capabilities of $dg(cmp)$, are known), or as in the traditional rating model, might not have access to any delegation knowledge except for $dctx(i, cmp) = ratee$, i.e. $\forall c \in C \setminus \{cmp\}, dctx(i, c) = \perp$.

4 Context Exploitation for Reputation Assessment

A basic abstraction of a number of existing reputation assessment models is a tuple (wt, rep) , as detailed below.

$wt : I \rightarrow WV$, is an interaction weighting function, governing the contribution of each available previous interaction $i \in I$ for the reputation assessment at hand. Here, WV is the set of possible weight values, and can be a binary domain, corresponding to an interaction selection decision, or a continuous domain, corresponding to an interaction ranking decision. Commonly, the factors playing a role in the evaluation of an interaction's weight $wt(i)$, are its recency $time(i)$ (recent observations are usually favoured over older ones), and the observing party $rater(i)$ (direct experience is usually favoured over third party opinions).

$rep : AN \cup \{\text{overall}\} \rightarrow PV$, is the reputation assessment function, providing a numeric value $rep(a) \in PV$ that reflects the level of trust the assessor places in the ratee (at the current time point) with respect to aspect $a \in attr(cmp) \cup \{\text{overall}\}$ of capability cmp . Depending on the reputation model and the domain of rating values, $rep(a)$ may refer to the expected probability of success, the expected rating, etc. It is usually estimated by applying some statistical summary function, **aggr**, over the ratings of interactions $i \in I$, while accounting for their weights $wt(i)$, i.e., $rep(a) = \mathbf{aggr}(\{ \langle wt(i), rating(i, a) \rangle \}_{i \in I})$. For example, **aggr** may correspond to a weighted mean over numeric ratings, or a probability estimation measure over categorical (e.g. binary) ratings, etc.

The extension we propose to existing reputation models is to account for an interactions's delegation context, $dctx(i, c \in C)$, in the assessment of its weight $wt(i)$. The idea is to assign higher weights to those interactions sharing similar delegation context with the potential interaction under consideration. The intuition behind this is straightforward: the best indication of how a provider may behave in future is how the provider behaved under *similar conditions* in the past, with differing conditions potentially leading to different behaviour. To achieve this, we are interested in quantifying the relevance between the delegation context of each existing interaction $i \in I$, and that of the potential future interaction, which we refer to as fi . Details are presented next.

4.1 Delegation Context Relevance Assessment

We aim for four criteria to be accounted for by the measure of relevance between the delegation contexts of interactions $i \in I$, and that of future interaction fi .

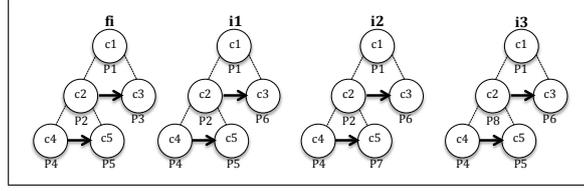


Fig. 2. Multi-level Relevance: i_1 is more relevant than i_2 and i_3

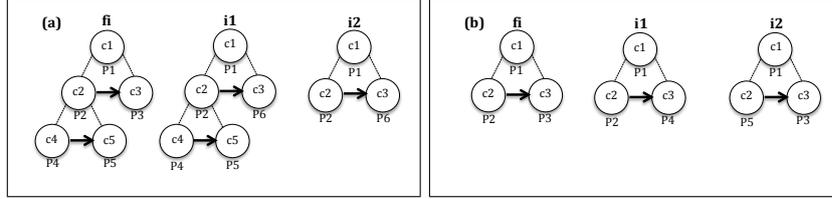


Fig. 3. Hidden Deleg. Context (a), Attr. centric Cap. (b): i_1 is more relevant than i_2

1. *Multi-level relevance*, taking into consideration similarities of delegation context across all levels of the delegation hierarchy. For example, in Figure 2, i_1 should be considered more relevant to fi than both i_2 and i_3 . This is because, when compared to i_1 , i_2 further differs from fi in the leaf provider of capability c_5 , while i_3 further differs in the intermediary provider of capability c_2 . In fact, whilst it is important to account for differences at the level of leaf sub-contractors since these are the ones actually performing the requested capabilities, intermediary sub-contractors may also play an important role in coordination, passing on truthful information, etc.
2. *Hidden delegation context*, discounting the relevance of an interaction in the case of uncertainty (i.e. missing information) regarding its delegation context. For example, in Figure 3(a), i_1 should be considered more relevant to fi than i_2 . This is because, unlike i_1 , i_2 's instantiation of capabilities c_4 and c_5 could be different from those of fi .
3. *Attribute-centric capabilities*, assigning higher importance to those capabilities in the delegation hierarchy that have an impact on the attribute under assessment. For instance, consider Figure 3(b) and assume that capabilities c_1 , c_2 , and c_3 correspond to *FHD*, *FPP*, and *FPD* of our motivating example, and that the attribute under assessment is *portion size* of the delivered food. Both i_1 and i_2 differ from fi in one provider, but i_1 should be considered more relevant to fi than i_2 . This is because, since food package delivery (*FPD*) has no effect on *portion size*, which is mainly determined by food package preparation (*FPP*), any difference in the performances of providers P_3 and P_4 would not cause any deviation for this attribute, as opposed to a difference between P_2 and P_5 , which could potentially affect *portion size*.
4. *Composition structure*, taking into consideration the various connectivity constructs (e.g. sequence, parallel, loop) among sub-capabilities in a decom-

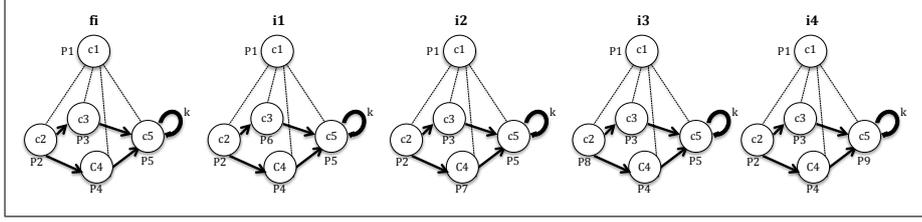


Fig. 4. Composition Structure: i_1 is more relevant than i_2 ; i_3 is more relevant than i_4

position graph. For instance, consider Figure 4, where capabilities c_3 and c_4 are performed in parallel, and c_5 is repeated k times. Assume assessment of *execution time*, with the domain knowledge indicating that c_4 usually takes much longer to be achieved than c_3 . Whilst both i_1 and i_2 differ from fi in one provider in the parallel construct, i_1 should be considered more relevant than i_2 to fi . This is because, given that c_3 's execution time is normally dominated by that of c_4 regardless of the provider, any difference in the performances of providers P_3 and P_6 is unlikely to affect the overall execution time observed for the interaction, as opposed to a similar difference in the performances of P_4 and P_7 . Similarly, i_3 should be considered more relevant to fi than i_4 since a difference in the performances of P_9 and P_5 would be magnified k times, unlike a similar difference between P_8 and P_2 .

A simple measure of relevance, $rel \in [0, 1]$, satisfactory of the above, between the delegation context of interaction i , $dctx(i, c \in C)$ (referred to as $dctx_i$ for simplicity), and the delegation context of future interaction fi , $dctx(fi, c \in C)$ (referred to as $dctx_{fi}$), with respect to attribute $a \in attr(cmp)$, can be given as:

$$rel(dctx_i, dctx_{fi}, a) = \sum_{\substack{c \in C, \\ dctx(fi, c) \neq \perp}} role(c, a) \times prel(dctx(i, c), dctx(fi, c)) \quad (1)$$

Function $role(c, a) \in [0, 1]$ defines the attribute-dependent distribution of roles among the capabilities under assessment (those *instantiated* with a provider in fi). That is, it specifies the relative importance of a capability for the relevance assessment, s.t. $\sum_{c \in C, dctx(fi, c) \neq \perp} role(c, a) = 1$. In particular, the role of capability c regarding attribute a is determined according to its hierarchical importance, W_h , structural importance, W_s , and attribute-related importance, W_a , as:

$$role(c, a) = \frac{W_h(c) \times W_s(c, a) \times W_a(c, a)}{\sum_{\substack{c_j \in C, \\ dctx(fi, c_j) \neq \perp}} W_h(c_j) \times W_s(c_j, a) \times W_a(c_j, a)} \quad (2)$$

The hierarchical importance of capability c , $W_h(c) \in [0, 1]$, is governed by c 's level in the *instantiated* delegation hierarchy of fi . Alternative options include:

(a) assigning equal importance to all levels, i.e. $W_h(c) = 1$ ($\forall c$); (b) favouring capabilities at lower levels, e.g. $W_h(c) = \frac{lvl(c)}{maxlvl(c)}$, where $lvl(c)$ is the level of c in the delegation hierarchy (with $lvl(cmp) = 1$), and $maxlvl(c)$ is the number of levels in fi 's longest *instantiated* hierarchy path containing capability c ; and (c) accounting only for the capabilities at the lowest level, i.e. $W_h(c) = 1$ if c is a leaf capability in fi 's *instantiated* delegation hierarchy, and $W_h(c) = 0$ otherwise.

The structural importance of capability c regarding attribute a , $W_s(c, a) \in \mathbb{Z}^+$, is governed by the position of c and its ancestors, $ancestor(c)$, in the corresponding decomposition graphs, and can be defined as follows:

$$W_s(c, a) = \prod_{c_j \in \{c\} \cup ancestor(c)} localW_s(c_j, a) \quad (3)$$

with $localW_s(c_j, a) = occur(c_j, unfold(critical(cg(c_j), a)))$. Here: $occur(c, g)$ is the number of occurrences of capability c in graph g ; $cg(c)$ is the decomposition graph containing capability c in the delegation hierarchy; $critical(g, a)$ returns the sub-graph of g that is considered critical for attribute a (i.e. the subgraph determining the performance regarding attribute a); and $unfold(g)$ returns the unfolded version of graph g using loop unfolding [14].

The attribute-related importance of c regarding attribute a , $W_a(c, a) \in [0, 1]$, can be defined as follows: $W_a(c, a) = 1$ if $a \in attr(c)$; and $W_a(c, a) = 0$, otherwise.

Finally, function $prel(dctx(i, c), dctx(fi, c)) \in [0, 1]$ measures the relevance between providers $dctx(i, c)$ and $dctx(fi, c)$, responsible for performing c in i and fi , respectively. It can be defined as follows:

$$prel(dctx(i, c), dctx(fi, c)) = \begin{cases} 1 & \text{if } dctx(i, c) = dctx(fi, c) \\ 0.5 & \text{if } (dctx(i, c) \neq dctx(fi, c)) \wedge (dctx(i, c) = \perp) \\ 0 & \text{if } (dctx(i, c) \neq dctx(fi, c)) \wedge (dctx(i, c) \neq \perp) \end{cases} \quad (4)$$

The correspondence between the criteria outlined earlier and the suggested relevance equation rel can be summarised as follows. The *multi-level relevance* factor is captured via the aggregation function (the sum function in Equation 1) over the scores of relevant capabilities, as well as via the hierarchical importance component W_h of function $role$. The *hidden delegation context* factor is captured via the provider relevance function $prel$, assigning lower relevance value (i.e. 0.5) in the case of missing (unavailable) instantiation of a capability. Finally, both the *attribute-centric capabilities* and the *composition structure* factors are captured in function $role$, via the attribute-related importance component W_a and the structural importance component W_s , respectively.

4.2 Reputation Model Extension

In this Section, we show how an existing reputation model, FIRE [4], can be extended to account for the delegation context relevance proposed in Equation 1. 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. We do not consider the role-based and certified reputation components in this paper.

Reputation is assessed in FIRE from tuples of form $(\alpha, \beta, a, i, rating(i, a))$, where α and β are agents that participated in interaction i such that α gave β a rating value of $rating(i, a) \in [-1, +1]$ for the term a . A rating of $+1$ is absolutely positive, -1 is absolutely negative, and 0 is neutral. To determine direct reputation of agent β for term a , an assessing agent α extracts the set of ratings from its database of the form $(\alpha, \beta, a, -, -)$ where “-” matches any value. Moreover, agents maintain a list of acquaintances, and use these to identify witnesses to evaluate witness reputation. Specifically, an evaluator α will ask its acquaintances for ratings of β for term a (i.e. ratings of the form $(-, \beta, a, -, -)$). Finally, the overall trust is calculated as a weighted mean of each of the component sources.

Since we focus on investigating the effect of delegation context on reputation, we do not consider the effect of the rating party or the interaction topology in this paper. That is, for simplicity, we assume that all previous interactions with agent β are accessible to any reputation assessor (e.g. via a dedicated rating repository), assigning equal importance to all raters. In FIRE, this corresponds to a fully connected agent network, with equal weights being assigned to the individual and witness experience. Thus, to determine the reputation of a provider P_{cmp} on behalf of a client, the assessor queries the rating store for ratings of the form $(-, P_{cmp}, a, i, rating(i, a))$. These ratings are scaled using a *recency factor*, λ , in the interaction weight function, instantiated in FIRE per interaction i as:

$$wt(i) = recency(i) = e^{\frac{|time(i) - time(fi)|}{\lambda}} \quad (5)$$

The reputation value the assessor has in P_{cmp} for term a is then calculated as the weighted mean of the available ratings: $rep(a) = \frac{\sum_{i \in I} wt(i) \times rating(i, a)}{\sum_{i \in I} wt(i)}$. Note that, to combine the reputation of different attributes a into a single composite assessment for agent P_{cmp} , we use a weighted sum across all attributes: $rep(overall) = \sum_a rep(a) \times attrwt(a)$, where $attrwt(a)$ corresponds to the weight of attribute a for the client, such that $\sum_a attrwt(a) = 1$.

Now, in order to account for the delegation context in FIRE, we adjust the weighting that is given to an interaction i so that it becomes attribute dependent and incorporates delegation context relevance, as follows:

$$wt(i, a) = recency(i) \times rel(dctx_i, dctx_{fi}, a) \quad (6)$$

5 Experiments and Results

This section presents an empirical evaluation of the proposed delegation-context-aware reputation framework, focusing on its performance in terms of producing more accurate reputation assessments⁶. The simulation involves one composite

⁶ Source code and data for the results presented in this paper are freely available from <http://jaspr.org/source-code>

provider agent interacting with a number of customers. In particular, we adopt our example scenario, showing the results from the perspective of the food home delivery provider P_{FHD} , with the delegation hierarchy of Figure 1. We assume that each (sub-)capability can be delegated to *pnm* alternative providers.

The simulation proceeds on the basis of rounds, each corresponding to an interaction between a customer and composite provider P_{FHD} . In each round, provider P_{FHD} instantiates its delegation hierarchy with a particular combination of sub-contractors, and accordingly delivers particular values for the aspects of interest. The customer then rates provider P_{FHD} on each aspect according to their satisfaction. The customer ratings and the provider's delegation context of the current round are utilised by the reputation assessor to adjust the reputation of provider P_{FHD} for the next round. Other experimental settings are outlined in Section 5.1 and Section 5.2, followed by experimental results in Section 5.3.

5.1 Customer Rating Generation

The evaluation considers three attributes: execution time (ex), quality of packaging (qp), and portion size (pz) of the delivered food. We assume that the performance of provider P_{FHD} with respect to each of these attributes is determined by the corresponding performances of the leaf sub-contractors who actually perform the capabilities (therefore we utilise option (c) for weights W_h).

Assuming knowledge that the food preparation (FPR) normally takes much longer than the packaging preparation (PPR), i.e. PPR does not belong to the critical path for evaluating ex , and that the packaging could be damaged during food packaging (FPK) or during food package delivery (FPD), the values delivered by provider P_{FHD} in an interaction for each considered attribute are:

$$\begin{aligned} val_{prv}(P_{FHD}, ex) &= val_{prv}(P_{FPR}, ex) + val_{prv}(P_{FPK}, ex) + val_{prv}(P_{FPD}, ex) \\ val_{prv}(P_{FHD}, qp) &= \min(val_{prv}(P_{PPR}, qp), val_{prv}(P_{FPK}, qp), val_{prv}(P_{FPD}, qp)) \\ val_{prv}(P_{FHD}, pz) &= val_{prv}(P_{FPR}, pz) \end{aligned}$$

Here, P_c denotes the provider selected for executing capability c in the interaction, and $val_{prv}(P_c, a)$ is the value produced by provider P_c for attribute a during the interaction. The generation of values val_{prv} for atomic providers is governed by their *attribute policies*, which are represented as normal distributions (with mean μ and variance σ^2) over the corresponding attribute domains, and assigned per candidate atomic provider at the beginning of experiments.

Based on this, the utility *perceived* by the customer in an interaction regarding aspect $a \in \{ex, qp, pz\}$ is: $utility_{prv}(a) = val_{prv}(P_{FHD}, a)$, while the utility considered *acceptable* by the customer is: $utility_{acc}(a) = val_{acc}(a)$, where $val_{acc}(a)$ corresponds to the value considered acceptable for attribute a (fixed among all customers in our experiments). Given this, the rating assigned by the customer for aspect a in interaction i , $rating(i, a) \in [-1, +1]$, compatible with the reputation model discussed in Section 4.2, equals to:

$$rating(i, a) = \begin{cases} \frac{utility_{prv}(a) - utility_{acc}(a)}{\max(a) - utility_{acc}(a)} & \text{if } utility_{prv}(a) \geq utility_{acc}(a) \\ \frac{utility_{prv}(a) - utility_{acc}(a)}{utility_{acc}(a) - \min(a)} & \text{otherwise} \end{cases}$$

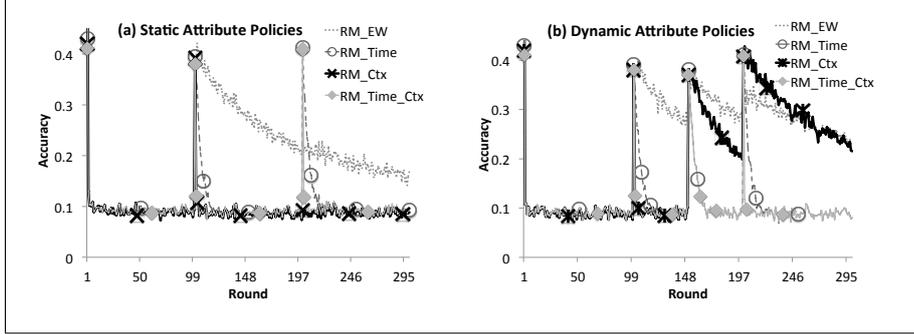


Fig. 5. Effect of Delegation Context Exploitation

where $\min(a)$ and $\max(a)$ are the minimum and maximum possible values for a .

Finally, the overall rating of provider P_{FHD} in interaction i is given as: $rating(i, overall) = \sum_{a \in \{ex, qp, pz\}} attrwt(a) \times rating(i, a)$, where $attrwt(a)$ is the weight of attribute a for the customer (fixed to $\frac{1}{3}$ for the three attributes).

5.2 Evaluation Strategies and Measure

We refer to the following reputation strategies: RM_EW , the reputation model assigning equal weights to all interactions, i.e. $\forall i, wt(i, a) = 1$; RM_Time , the reputation model weighting interactions according to recency, i.e. according to Equation 5; RM_Ctx , the reputation model weighting interactions according to delegation context relevance, i.e. according to Equation 6 with $\forall i, recency(i) = 1$; and RM_Time_Ctx , the reputation model weighting interactions according to both recency and delegation context relevance, i.e. according to Equation 6. As a performance measure, we quantify the difference between the provider's reputation exposed to the customer prior to an interaction i , $rep^i(a)$ (which, given the reputation model adopted, can be viewed as the predicted rating for the interaction), and the customer's actual rating following the interaction, $rating(i, a)$. That is, we measure $|rep^i(a) - rating(i, a)|$ at each round (with $a = overall$).

5.3 Results

In this section, we compare the outlined strategies under various environment settings. All the results reported are averaged over 100 simulation runs.

The effect of delegation context exploitation. Figure 5(a) reports the results in settings with *dynamic* delegation context, and assuming *static* attribute policies of sub-providers. In particular, composite provider P_{FHD} changes its delegation context (i.e. switches its sub-providers to different ones) after 100 rounds, then returns back again to the old delegation context after another 100 rounds. As can be seen, RM_EW suffers from poor accuracy after the first change in the delegation context, since the reputation score mostly reflects old, no longer relevant ratings. This degradation in performance is of less severity after the second

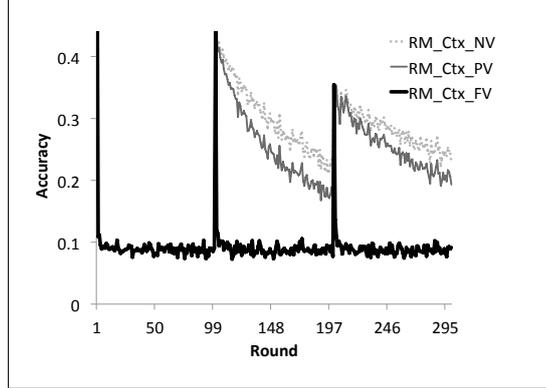


Fig. 6. Effect of Visibility Levels of Delegation Context

(recurring) change, with the ratings observed during the first 100 rounds becoming relevant again. Strategy *RM.Time* achieves better adaptive behaviour by favouring more recent ratings and gradually forgetting outdated ones, but is outperformed by *RM.Ctx* (the best performing strategy in this case). This is because, by utilising knowledge of delegation context, *RM.Ctx* incorporates only the most relevant ratings into reputation assessment (eliminating irrelevant ratings, collected under different delegation context). Thus, it achieves the fastest recovery of accuracy after the first change, and avoids an accuracy drop after the second change (by favouring the ratings collected in the first 100 rounds, which become relevant again). No further performance improvement is achieved by combining delegation context awareness with recency in this case.

Figure 5(b) reports the results with *dynamic* delegation context settings as above, but also with *dynamic* attribute policies of sub-providers. In particular, the attribute policies of each candidate sub-provider are set to change after 150 rounds, with a policy change being simulated by a repositioning of the corresponding mean μ . Here, although *RM.Ctx* still achieves dominating results after the change of the delegation context at round 100, it fails to do so once the providers' policies change at round 150, and further deteriorates in performance when the old delegation context reoccurs at round 200. This is because, *RM.Ctx* considers all the ratings collected under similar delegation context to be of equal importance, despite the fact that those collected prior to round 150 may no longer be relevant. The best performing strategy in this case is *RM.Time.Ctx*, which can eliminate the effect of such irrelevant ratings with time, while keeping the advantages of delegation context utilisation.

The effect of delegation knowledge granularity. Figure 6 compares the performance of *RM.Ctx* under various delegation context visibility levels, in settings with *dynamic* delegation context as above and *static* attribute policies of sub-providers. In particular, we compare: *RM.Ctx.FV*, assuming full visibility of the delegation context; *RM.Ctx.PV*, assuming partial visibility of the delegation context, where only the direct sub-contractors, P_{FPP} and P_{FPD} , of compos-

ite provider P_{FHD} are known; and RM_Ctx_NV , assuming no visibility of the delegation context, i.e. the only provider visible is provider P_{FHD} . Changes in delegation context are assumed to only affect leaf capabilities FPR , PPR , FPK , and FPD , while the intermediary provider of capability FPP always remains the same. Clearly, RM_Ctx_NV does not detect delegation context changes, assigning equal weights to all ratings for the duration of the simulation, thus exhibiting bad performance after change points. RM_Ctx_PV , on the other hand, is only able to observe the change in the sub-provider of leaf capability FPD . As a result, it discounts the importance of the ratings before the change, but does not eliminate their effect entirely (these ratings are still considered partially relevant), which decreases its accuracy compared to RM_Ctx_FV .

6 Discussion

Why would providers expose (true) delegation context? Providers are the obvious source of delegation context as it is a record of how they provided a service, but it may be against their interests to release such records. There are a few initial answers to this question, though full exploration of the issue is beyond the scope of this paper. First, such information should be expected to be present in the client-accessible service advert at the time of service provision. Second, there are two agents in an interaction that could provide (or verify) information regarding a particular delegation, the delegator and the delegatee (a commonly used mechanism for non-repudiation). Finally, the contracts which clients agree with providers can require some recording of details as part of service provision, possibly with involvement of a notary to help ensure validity.

How would providers expose delegation context? The PROV standard [10] (published by W3C as a standard for interoperable provenance) could provide a suitable solution for this purpose [15]. A PROV document describes in a queryable form the causes and effects within a particular past process of a system (such as agents interacting, the execution of a program, or enactment of a physical world process), as a directed graph with annotations. The contents of a provenance graph can be collated from data recorded by a set of independent agents, and clients have a standard means to query the data, e.g. by SPARQL⁷.

7 Related Work

In a service-oriented system individuals and organisations rely on providers to successfully execute services with an appropriate quality to fulfil their own goals, and such reliance implies a degree of risk. Trust and reputation provide an effective way of assessing and managing this risk, and are studied by researchers from many domains. In multi-agent systems, most established computational reputation models, such as TRAVOS [1], HABIT [2], ReGreT [3] and FIRE [4], typically use a combination of direct and indirect experience. In TRAVOS [1],

⁷ <http://www.w3.org/TR/sparql11-overview/>

the trust score is the expected probability (using the beta distribution) that the trustee will fulfil its obligations towards the trustor in an interaction, estimated based on the outcomes of the previous direct interactions with the trustee. When there is a lack of personal experience, the trustor seeks the opinions of other sources, accounting for their reliability. Similarly, HABIT [2] uses a probabilistic approach, utilising Bayesian network to support reasoning about reputation. ReGreT [3] takes into account three dimensions of reputation: the individual dimension (based on direct interactions), the social dimension (from other sources utilising the group relation), and the ontological dimension (defining the different reputational aspects). FIRE [4] (adopted in this paper) is based on ReGreT, adding role-based trust and certified reputation based on third-party references.

Trust and reputation models have also been investigated in service-oriented systems. For example, Maximilien et al. [5] estimate a service’s reputation for a quality by aggregating its previously observed quality values (shared and accessible to all assessors). Similarly, Xu et al. [6] extend the UDDI registry with a reputation manager, aggregating the past ratings of a service into a reputation score. Malik et al. [7] propose a decentralised approach for service reputation assessment, where customers seek ratings from their peers, with the credibility of ratings being estimated based on deviation from the majority opinion.

These approaches view a service as a simple service, ignoring potential composition information behind its provision, and relying mainly on recency to discount the effect of irrelevant past interactions. We argue that recency alone is not sufficient as the behaviour of a composite service is affected by its underlying composition circumstances, which may change, or older ones reoccur (making older interactions better predictors of the current service behaviour). To provide more accurate indications of interaction relevance, our work complements such recency-based interaction weighting with composition-context-based weighting.

Finally, a number of researchers focus on designing suitable mechanisms for distributing the score obtained by a composite service to its component services [8, 9]. Proposed factors for governing such distribution include a component service’s structural importance, replaceability, and run-time performance. However, these approaches do not account for the implications of the component services on the reputation evaluation mechanism of the composite service itself (the focus of this paper), but can be considered complementary to our work.

8 Conclusion

This paper presented how delegation information underlying a composite service provision can be utilised to provide more accurate reputation assessment of the composite service provider. Specifically, such information is used to scale the ratings available for the provider, assigning higher weights to those collected under circumstances comparable to the current settings. The proposed composition-context-based weighting is independent of any particular reputation model, but for evaluation purposes, was incorporated into an existing reputation model, FIRE, which scales ratings according to recency. The results show that it re-

sults in improving performance. Future work involves accounting for alternative decomposition graphs per capability, and for personalised user requirements.

Acknowledgments. 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. Teacy, W. T. L., Patel, J., Jennings, N. R., Luck, M.: Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In: 4th Int. Conf. on Autonomous Agents and Multiagent Systems, pp. 997–1004 (2005)
2. Teacy, W. T. L., Luck, M., Rogers, A., Jennings, N. R.: An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*. 193, 149–185 (2012)
3. Sabater-Mir, J., Sierra, C.: Regret: A reputation model in gregarious societies. In: 4th Workshop on Deception, Fraud and Trust in Agent Societies, pp. 61–69 (2001)
4. Huynh, T. D., Jennings, N. R., Shadbolt, N. R.: An integrated trust and reputation model for open multi-agent systems. *J. of Autonomous Agents and Multi-Agent Systems*. 13, 119–154 (2006)
5. Maximilien, E. M., Singh, M. P.: Agent-based Trust Model Involving Multiple Qualities. In: 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems, pp. 519–526 (2005)
6. Xu, Z., Martin, P., Powley, W., Zulkernine, F.: Reputation-Enhanced QoS-based Web Services Discovery. In: IEEE Int. Conf. on Web Services, pp. 249–256 (2007)
7. Malik, Z., Bouguettaya, A.: RATEWeb: Reputation Assessment for Trust Establishment Among Web Services. *The VLDB Journal*. 18, 885–911 (2009)
8. Nepal, S., Malik, Z., Bouguettaya, A.: Reputation Propagation in Composite Services. In: IEEE Int. Conf. on Web Services, pp. 295–302 (2009)
9. Wen, S., Li, Q., Yue, L., Liu, A., Tang, C., Zhong, F.: CRP: context-based reputation propagation in services composition. *Service Oriented Computing and Applications*. 6, 231–248 (2012)
10. W3C. PROV model primer. <http://www.w3.org/TR/prov-primer/>, 2013.
11. Pinyol, I., Sabater-Mir, J.: Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*. 40, 1–25 (2013)
12. Gambetta, D.: Can we Trust Trust?. In: *Trust: Making and Breaking Cooperative Relations*. Ed. by D Gambetta, pp. 213–237 (1988)
13. Jsang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*. 43, 618–644 (2007)
14. Zeng, L., Benatallah, B., Ngu, A. H. H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware Middleware for Web Services Composition. *IEEE Trans. Softw. Eng.* 30, 311–327 (2004)
15. Miles, S., Griffiths, N.: Incorporating Mitigating Circumstances into Reputation Assessment. In: 2nd Int. Workshop on Multiagent Foundations of Social Computing (2015)
16. Smith, R. G., Davis, R.: Frameworks for Cooperation in Distributed Problem Solving. *IEEE Transactions on Systems, Man, and Cybernetics*. 11, 61–70 (1981)