A Personalized Approach to Experience-Aware Service Ranking and Selection

Friederike Klan and Birgitta König-Ries

Institute of Computer Science, Friedrich-Schiller-University Jena, Ernst-Abbe-Platz 1-4, 07743 Jena, Germany
{friederike.klan,koenig}@informatik.uni-jena.de

Abstract. Existing approaches to service ranking and selection evaluate the suitability of available services for a given request based on the advertisement created by the service provider. They will compare how well the advertisement matches the service request and will choose the service with the best matching advertisement. Unfortunately, at this point in time, it is uncertain whether the service that will actually be performed will match the request as well as the advertisement promised. In this paper, we present an approach that reduces the degree of this uncertainty by taking previous experiences with the service provider (which reflect the performance of the actual service *not* the advertisement) into account. Contrary to many other approaches our solution accounts for the subjective nature of rating-based experiences by considering the preferences of the experience creators. Moreover it exploits the number of available experiences more effectively by considering not only experiences for a given service, but also experiences for similar services of the same provider. Our solution utilizes indirect user information and avoids explicit sharing of personal consumer information.

1 Introduction

Over the last decade the Web evolved from a collection of static web sites offered by a relatively small number of providers to a platform for sharing and collaboration where everyone can provide content and offer functionality. This shift of web usage behavior and the consequent rise of available information and resources as well as their growing heterogeneity poses new challenges to application integration. A paradigm that has proved to be appropriate to enable this is service-oriented computing. It allows to provide functionality or information as stand-alone services, that can be described by a service offer, then published, automatically discovered, and ranked by comparing (matching) a given service request with available offers, selected, composed and executed. Facing the overwhelming flood of information effective mechanisms for matching and ranking available resources according to their relevance become crucial to support consumers when selecting a service. In this context much work has been done on improving the expressiveness of service descriptions, particularly by applying semantic techniques, as well as on effective matchmaking algorithms to select

S. Greco and T. Lukasiewicz (Eds.): SUM 2008, LNAI 5291, pp. 270–283, 2008.

suitable service offers based on a given request. However, there is a critical point here that has not been paid much attention to so far. Existing matchmaking approaches implicitly assume that the service advertised in the offer corresponds to the actually provided service. However, it is unrealistic to assume to have a 1:1 correspondence here. In fact one cannot be sure about the degree of correspondence between offer and actual service. This is due to several reasons. On the one hand service providers may advertise more than they are able to provide to be preferred over competing providers. On the other hand discrepancies between the promised and the actually provided service naturally arise from the dynamics of services as well as from the finiteness of service descriptions. In general, there is a tradeoff between the accuracy and the size of a service description and thus the cost of matchmaking. Thus service descriptions tend to be inaccurate to some degree [1]. Though this problem can be mitigated by introducing a negotiation step it cannot be solved completely. It is obvious that the bigger the difference between the advertised and the actually provided service the less meaningful are the results of the service matchmaker and the more arbitrary is the offer ranking and thus the selection decision based on those results, since available services are solely compared on the basis of their offer descriptions. Service ranking and selection algorithms should take this point into consideration by 1) quantifying the difference between the offer and the actual service (*offer conformance*) and thus the preciseness of the matching results and 2) considering this information when ranking available services.

In this context collaborative feedback mechanisms seem to be promising to reduce the uncertainty in service selection [2]. However existing mechanisms for experience-aware service selection exhibit a number of disadvantages. In most of the approaches offer conformance is a rather abstract concept [2], neither related to the attributes of a service nor to the service promised in the offer. A specific class of approaches in that context are those that recommend suitable services based on selections of other users [3,4]. In our opinion these approaches suffer from a major drawback: they do not consider whether the decision of the user was a good one or not, i.e., they record user satisfaction with the offer *not* with the actual service delivered. Often solutions that measure the performance of a service in terms of its attributes captured in the offer description assume consumer feedback to be objective and measurable [5,6,7,8,9]. Thus, they mainly focus on QoS attributes [10,8,9]. Though some service properties' values can be measured automatically, most of them, particularly in the field of information services, cannot. Imagine for instance a service that provides digital contents like a mp3-file containing a song. In that case one can automatically verify that the downloaded file is an mp3-file, but automatically checking that it contains the song you wanted by the interpreter you wanted is not possible. Moreover, measuring a service's performance with respect to several aspects can be very costly and has to be based on commonly agreed upon measuring methods. Due to these facts we advocate personalized experiences in terms of consumer ratings. Ratings are personalized, i.e they depend on a consumer's expectations expressed in his preferences. These preferences capture how important certain aspects of a service are to a consumer. Consequently experiences of specific consumers are only transferable to situations where the involved service consumer has similar preferences. Most of the existing rating based approaches [10,11,12] do not account for that fact. Some solutions, e.g. [7], that consider the subjective nature of ratings rely on the explicit exchange of consumer requests and preferences. In our opinion this would divulge valuable personal information. Moreover comparing user requests and preferences directly would be very costly. Consumer ratings for a specific service are scarce. For that reason feedback should be exploited effectively. Only a few approaches consider this [5,10].

In this paper we propose an approach for experience-aware service ranking and selection that solves the described problems. It is designed as an extension for existing matchmakers and utilizes consumer experiences in former service interactions to allow for proper service ranking and selection. The remainder of the paper is organized as follows. First we provide definitions for basic notions and specify our assumptions about the underlying service description language as well as on the underlying matchmaker (Sect. 2). Afterwards we present our solution for experience-aware service selection in Sect. 3. We discuss and present experimental results in Sect. 4 and conclude the paper in Sect. 5.

2 Prerequisites

Before describing the entire approach in the following sections we define basic notions and make some general assumptions about the underlying service description language as well as on the underlying *basic matchmaker*. We consider a *service* as a set of instances that can be executed. It is characterized by a set of attributes. Each *service instance* is characterized by a particular combination of values for those attributes. Service descriptions are set-based, i.e. they describe a service by means of its instances.[1] A *service offer or advertisement* specifies which instances a given service provides. For instance a service offer for a bookseller might describe that this offer contains the set of service instances where the effect is that ownership of a book changes. These books have a title, prices and so on. An example of a service instance would be the service selling the book with the title "Pope Joan" for 14.90 Euro, to be delivered by 01.04.08 to a certain address. A *service request* characterizes a *service consumer's* goal by describing the service instances that are suitable for solving that goal. Additionally, a service request contains an implicit mapping that assigns a *match value* to each service instance in the request. This value indicates how well a given instance fits to the service consumer's goal. W.l.o.g. we assume that the match value is a real number from the interval $[0, 1]$. The mapping can be interpreted as the *consumer's preferences* with respect to the given request. A service request could be the set of instances where the effect is that ownership

[1] We assume this set to be finite. Though this is not necessarily true for real service descriptions, we argue that an infinite instance set can be sufficiently approximated by a finite instance set.

of a book with the title "Pope Joan" changes to me, where the delivery address is my address, the price is less than 20 Euro and delivery happens within the next 2 days. We do not set any further restrictions on the underlying service description language. We assume that the basic matcher implements a pessimistic set-based matchmaking approach. Specifically that means that an offer matches to a given request, if it is a subset of the request. The *match value* of an offer and a request is the smallest match value of the service instances described in the offer according to the request. It indicates how well the considered offer fits to the request. The pessimistic approach ensures that the executed service instance has a match value that is at least as high as the match result for the whole service. This is reasonable, since we will not know in advance which of the service instances described in the offer is executed by the service provider. Once available offers are discovered the matcher determines a sorted vector containing the match values for the offers according to a given request (*match result vector*) in a completely automatic fashion. The best fitting offer, i.e. the offer yielding the highest match value, is chosen and the corresponding service is invoked, i.e. one of its instances is executed, without requiring additional human intervention. Among other approaches the semantic service description language DSD [13,14] and the DIANE service matcher meet these assumptions.

3 Experience-Aware Service Selection

Our solution is designed as an extension for existing matchmakers, i.e. we assume to have the matching results provided by a basic matcher based on a given request and available offer descriptions. To allow for proper service ranking we evaluate how reliable those results are. More precisely we have to quantify the degree of conformance between the offer, the matching result is based on, and the service that will be actually provided (*offer conformance*). For that reason each service consumer provides the offer conformance of a provider perceived during service interaction as experience to others. Obviously, experiences of users that had similar requirements are more valuable in this context. If, for instance, two users requested a mp3-file but with different encoding and bit rates, and one of them is dissatisfied with the service, that does not imply that the other will be, too. The offer conformance of a specific service provider with respect to a given request and thus the reliability of the matching results for his services can then be predicted based on *relevant experiences* provided by other consumers. Finally, available offers are ranked based on their match value provided by the underlying matcher, the offer conformance prediction and the confidence of that prediction. In this section, we will describe how offer conformance can be formalized (Subsect. 3.1), how it can be used to predict the future offer conformance of a provider (Subsect. 3.3), how relevance of experiences can be determined (Subsect. 3.2) and how services can be ranked based on this information (Subsect. 3.3). Algorithm 1 is the resulting overall algorithm.

Algorithm 1. ExperienceAwareServiceSelection(request r, offer set O)

1. get the matching results mr for the request r and the offers in O from the basic matcher;
2. drop all offers with a match value of 0;
3. **if** (mr contains more than one offer of the same provider) **then**
4. keep the offer with the highest match value among them and drop all the others;
5. **end if**
6. **for** (each provider with an offer in mr) **do**
7. get all available experiences that are relevant for predicting the offer conformance of that provider with respect to r;
8. calculate the overall offer conformance value and its confidence;
9. **end for**
10. rank the available offers based on mr, their predicted offer conformance and its confidence;
11. **return** t he offer with the highest rank;

3.1 Offer Conformance

Inspired by the concept of quality conformance proposed by Vu et al. [8], we measure the conformance between the service promised and that actually provided by comparing the match value calculated by the basic matcher and the rating the consumer provides after service execution. More specifically consider a consumer c posing a request r. Once the best fitting offer o among the set of available offers is determined, the corresponding service s is invoked. After service execution the consumer rates the service in terms of its suitability for reaching the goals captured in the request. Given the consumer c and the request r we measure the conformance $oc \in [0, 1]$ of the offer o and the service s actually provided by

$$oc(c, s, r, o) = \begin{cases} \frac{ra(c,s,r)}{mv(r,o)} & \text{if } mv(r, o) \geq ra(c, s, r) \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where $mv(r, o)$ is the match value for r and o provided by the basic matcher. The value $ra(c, s, r) \in [0, 1]$ is the personalized rating for the service s provided by the consumer c (with respect to the request r). It is the higher the more the service outcome suited to the consumer's needs. Note that we consider only negative deviations from the match value.

3.2 Relevance of Experiences

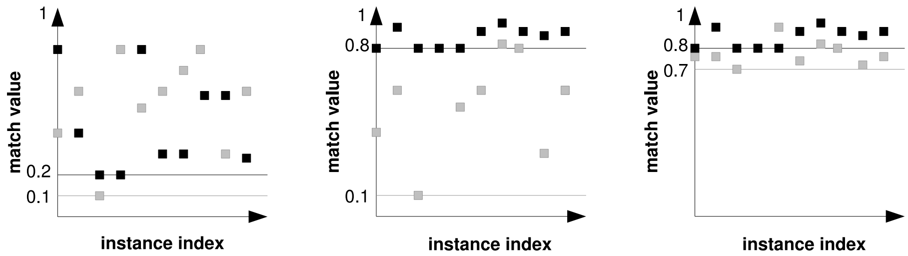
Once suitable offers and their providers are determined by the basic matcher, we have to identify consumer experiences that are *relevant* for predicting the offer conformance of every single provider with respect to the specified request. We define the set of experiences that are relevant for the offer conformance prediction of a given provider p with respect to the request r as those experiences that refer to a service stemming from p. An experience is the more relevant the more similar r and the request the experience is based on are. The first postulation is due to the fact that experiences with one provider cannot be transferred to another, the second accounts for the fact that ratings and thus offer conformance experiences are personalized and request-specific, i.e. experiences based on similar service demands and similar preferences are more valuable in this context. More specifically this means that observing the same service instance, the considered consumer and the experience provider should produce similar offer conformance values. This is true, if the given requests are similar. Moreover it is required that given similar requests the matcher provides similar match values. Presuming that the similarity of requests implies similarity of all their instances, this assumption holds if we choose $1-$ the mean deviation of corresponding service instances' match values (instances not contained in one of the requests were assumed to have a match value of 0) as a measure for request similarity. The relevance of a given experience may then be determined by directly comparing the two requests involved. However this is not a satisfying solution. The reasons are twofold. On the one hand the computation would be very costly, on the other hand consumers had to divulge much of their personal information including their preferences to allow for that comparison. Due to these facts we propose to approximate the similarity of the requests by indirectly comparing their match result vectors containing the match values for several offers. Note that this will not raise additional effort, since the match result vectors are created by the basic matchmaker anyway.

Approximating the request similarity. Without loss of generality we assume that both requests r_1 and r_2 are matched against the same set of offers,[2] i.e. all offers, and both resulting match result vectors mr_1 and mr_2 are in the same order with respect to the offers. The algorithm is based on two assumptions:

1. The result of the comparison between two corresponding match values $mr_1[i]$ and $mr_2[i]$, $0 \leq i \leq n$, where $n+1$ is the length of mr_1 resp. mr_2, provides an indication for request similarity/dissimilarity, if at least one of the corresponding match values is high. It is the more significant, the higher $\max(mr_1[i], mr_2[i])$ is.
2. If the result of a comparison is significant, the following holds: The smaller the differences between the corresponding match values, the higher the similarity of the corresponding requests.

Figure 1 illustrates the motivation for those assumptions. It shows three diagrams each indicating the service instances of a sample offer with 10 instances and their match values with respect to two different requests (black and gray points). The lines correspond to the match value of the overall offer with respect to the requests. Diagram 1(a) presents the case where the offer has a low match value with respect to both requests, 0.1 and 0.2 respectively. In that case the difference between the match values of a single service instance with respect to the two requests is between 0 and $1-0.1=0.9$, so we cannot draw any conclusions about the similarity of the requests. Diagram 1(b) shows the case where the offer has a high match value with respect to one request and a low match value with

[2] Note that this is a simplified assumption, since existing matchmakers often do not match a given offer but a request-specific configuration of an offer (specialization of the offer). Consequently we do not necessarily compare the match values for the same offers. However our algorithm will provide meaningful results, if the available offers are not too generic. In this case the configured offers do not differ that much.



(a) (b) (c)

Fig. 1. Motivation for the assumptions

Algorithm 2. ApproximateSimilarity(minimalMV,threshold,mr_1,mr_2)

```
sumOfDeviations = 0;
sumOfWeights = 0;
for (i < mr_1.length) do
  maximum = max(mr_1[i], mr_2[i]);
  if (maximum ≥ minimalMV) then
    maximum = maximum^2;
    sumOfDeviations+=| mr_1[i] − mr_2[i] | · maximum;
    sumOfWeights+=maximum;
  end if
end for
if (sumOfWeights ≥ threshold) and (sumOfWeights > 0) then
  return 1-(sumOfDeviations/sumOfWeights);
else
  return 0;
end if
```

respect to the other. In that case the difference between the match values of a single service instance with respect to the two requests is also between 0 and 0.9, but we presume that it is most often high, since we believe that in real requests the match values of the described service instances are similar to some degree. Thus case 1(b) is an indication for the dissimilarity of the two requests. Diagram 1(c) depicts the case where the offer has a high match value with respect to both requests. In that case the difference between the match values of a single service instance with respect to the two requests is between 0 and $1 - 0.7 = 0.3$. Thus we have a strong indication for the similarity of the two requests. Based on the above assumptions we calculate the approximate similarity sim_{approx} of two requests r_1 and r_2 according to Alg. 2 by comparing their corresponding match values for a set of comparison offers given by the match result vectors mr_1 and mr_2. The algorithm calculates the weighted mean of the match value deviations. Following Assumption 1 the weights $\max^2(mv_1, mv_2)$ correspond to the significance of each comparison result. Moreover for calculating the approximate similarity only those pairs are considered where at least one of the match values exceeds the value `minimalMV`. The output of the algorithm is only meaningful, if it is based on a sufficient number of significant comparisons. This is true, if the sum of the weights exceeds a given `threshold`, otherwise the algorithm returns 0, which means that the requests are not similar. Consequently the corresponding experiences are not considered for the offer conformance prediction. Appropriate values for $minimalMV$ and $threshold$ are application-specific and thus parameters of the algorithm. They can be determined based on test data (see Sect. 4).

Quality of the measure. The quality of our approximate similarity measure depends on the diversity and the number of available offers. This is due to the fact that we need a sufficient number of offers that lead to a high match value for at least one of the two requests to be compared to have enough match results that provide an indication for request similarity/dissimilarity. Since each single match value comparison allows to draw conclusions only about the request similarity referring to the service instances contained in the offer, a sufficient number of diverse offers is needed. On the other hand if a subset of the service instances covered by a request is not contained in any of the available offers, the quality of the similarity approximation with respect to those instances would be bad, but it does not matter, because those instances are never executed and thus never rated. However we argue that the algorithm provides meaningful results in real world scenarios, since in those settings we have a variety of offers. The quality of the approximate similarity measure will be evaluated in Sect. 4.

3.3 Service Ranking

Having determined the set of relevant experiences $E_{rel}(p, r)$ for each provider p with respect to the specified request r, we predict its future offer conformance $oc_p(r)$ based on those experiences. Moreover we calculate a confidence value $conf(oc_p(r))$ indicating the reliability of the offer conformance prediction $oc_p(r)$. Afterwards we rank the offers in the match result vector based on those values and the initial match values provided by the basic matcher.

Predicting the offer conformance. The future offer conformance $oc_p(r) \in [0, 1]$ for p's offers with respect to the specified request r is calculated as the weighted mean of the observed oc-values (see Equ. 1) provided by the experiences in $E_{rel}(p, r)$.

$$oc_p(r) = \begin{cases} \frac{\sum_{i \in E_{rel}(p,r)} w(i) \cdot oc(i)}{\sum_{i \in E_{rel}(p,r)} w(i)} & \text{if } E_{rel}(p, r) \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

with

$$w(i) = w_a(age(i)) \cdot sim_{approx}(r_i, r), \tag{3}$$

where $w(i)$ is a weight indicating the relevance of an experience i for the given request r. An experience is the more relevant the smaller its age $age(i)$ and the higher the similarity $sim_{approx}(r_i, r)$ of r_i and the considered service request r. The weight w_a with $w_a(a_{max}) = 0$ and $w_a(0) = 1$ is a monotonically decreasing function taking into account the age of an experience. Experiences older than a_{max} are not considered.

Confidence calculation. The confidence $conf(oc_p(r))$ of the predicted offer conformance value $oc_p(r)$ is determined by the number, age, and relevance of the experiences its calculation is based on. It is the higher the higher the number of experiences and the newer and the more relevant the experiences are. The confidence $conf(oc_p(r))$ of the offer conformance prediction $oc_p(r)$ is

$$conf(oc_p(r)) = \begin{cases} f(|E_{rel}(p,r)|, \min_{i \in E_{rel}(p,r)} w(i)) & \text{if } E_{rel}(p,r) \neq \emptyset \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where f is an application-specific function to the interval $[0, 1]$. It increases with the number of experiences and the minimal weight $w(i)$ of the experiences $i \in E_{rel}(p, r)$. It can be determined based on test data in the given field of application. We may also consider weights that account for the trustworthiness of the experience providers. However assessing the trustworthiness of experiences is a challenging topic in itself. It is not considered in this paper.

Service ranking. Once the $oc_p(r)$ and $conf(oc_p(r))$ values are calculated for the providers, we rank the offers based on those values and the initial match value of each offer. The rank $rank(o_p) \in [0, 1]$ of an offer o_p from the provider p with the match value $mv(r, o_p)$ is calculated by

$$rank(o_p) = mv(r, o_p) \cdot conf(oc_p(r)) \cdot (oc_p(r)^2 - 1) + mv(r, o_p). \quad (5)$$

The rank of an offer o_p is equal to $mv(r, o_p)$ as long as no experiences are available and decreases linear to $mv(r, o_p) \cdot oc_p(r)^2$ with increasing confidence. Intuitively spoken this means that unknown services are preferred to give newcomers a chance. On the other hand inaccurate description providers are punished, where the penalty is the higher the lower the offer conformance is. Algorithm 1 summarizes the overall service ranking and selection procedure. Given a request r it ranks the available offers O based on the match result vector provided by the basic matcher and available experiences.

4 Evaluation and Discussion

We implemented the service model and a basic matcher as described in Sect. 2 as well as our approach to experience-aware service selection introduced in this paper. We performed several simulative experiments to evaluate the effectiveness of our solution.[3]

In a first series of tests we evaluated the quality of our approximate similarity measure, in a second series we investigated the quality of the offer conformance prediction. In both cases we studied the dependency of the quality from several parameters.

[3] The simulation environment as well as the implementation of the tests described in this section are available under http://fusion.cs.uni-jena.de/professur/?content=fklan.

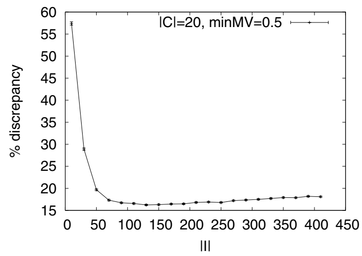
Algorithm 3. TestApproxSim($|I|$,$|C|$,$minMV$)

1. create two random requests of randomly chosen similarity containing maximal $|I|$ instances
2. create $|C|$ random comparison offers having at least a match value of $minMV$ with the first request
3. calculate approximate and actual similarity for the requests as described in Sect. 3.2
4. **return** t he percental deviation of the approximate similarity from the actual similarity

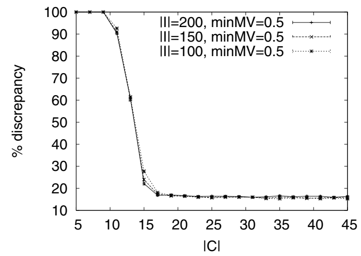
Quality of the approximate similarity measure. We evaluated the quality of our approximate similarity measure (Alg. 2) depending on the maximum number of instances per request $|I|$, the cardinality of the comparison set of offers $|C|$, i.e. the number of offers considered in the match result vectors which are input to the approximate similarity algorithm, and the match values of the offers in that set. For the latter we introduced a parameter $minMV$ indicating the smallest match value of all offers in the comparison set with respect to a given requests. The quality was measured in terms of the percental deviation of the approximate measure's results from the actual similarity calculated by comparing requests directly as described in Sect. 3.2.

The test was performed according to Alg. 3. It was run with several parameter settings. The average quality was calculated over 10000 runs (50000 for the quality-$|I|$-dependency plot). We tested our approximate similarity measure with several parameter configurations. We identified $threshold = 10$ and $minimalMV = 0.5$ as a good combination. Decreasing one or both of the values leads to lower quality results. Increasing the parameter values does not result in a significant improvement of the result's quality. All test series are based on this parameter setting. Figure 2(a) shows the quality of the approximate similarity measure depending on the maximum number of instances per request.[4] After an initial phase the average deviation of the approximated similarity is about 17% for $|I| \approx 100$ and slightly increases with higher $|I|$-values. Another series of experiments showed that the absolute deviation for $|I| \approx 100$ is 0.1. Further increasing of $|I|$ up to 5000 instances results in a deviation of about 20%. The bad quality for small request lengths originates from the small number of possible offers in the comparison set. We evaluated the quality of the approximate similarity measure depending on the minimal match value of the offers in the comparison set with respect to the given request. Resulting from the higher match values of the comparison offers and thus a higher significance of the single comparison results, the quality increases with higher values for $minMV$ and reaches its optimum around $minMV = 0.5$. Further increasing of $minMV$ leads to results of worse quality. This is due to the decreasing number of possible comparison offers. The dependency between the quality of the approximate similarity measure and the cardinality of the set of comparison offers is illustrated in Fig.2(b). The quality is plotted for 3 different values of $|I|$. After an initial phase the quality remains stable at a level of about 17% for $|C| = 20$. Further increasing of $|C|$ does not result in a better quality. Summarizing the results for this test series we point out that a number of 20 comparison offers with a match value of at least 0.5

[4] The brackets indicate the standard error of the sample mean assuming that the discrepancy values follow a Gaussian distribution.



(a)



(b)

Fig. 2. Quality of the approximate similarity measure depending from the maximal number of instances per request (a) and the cardinality of the comparison set (b)

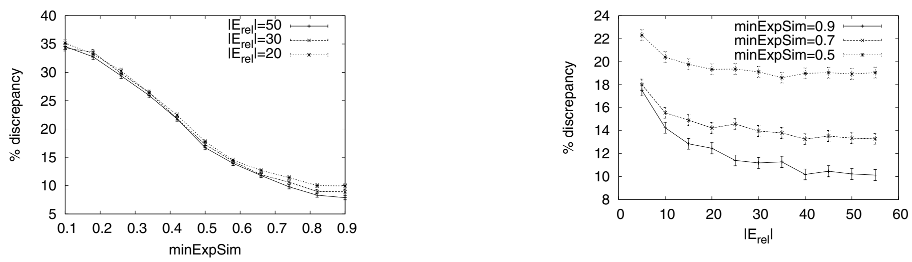
with respect to at least one of the two requests to be compared is sufficient to assure a measure quality of about 17%. The quality just slightly decreases with the maximal number of instances per request.

Quality of the offer conformance prediction. The quality of the offer conformance prediction was evaluated depending on the number of relevant experiences $|E_{rel}|$ and the minimal similarity $minExpSim$ between the given request and those the experiences are based on. We simplified our experiments in that we considered only service providers offering a single service. We did not consider providers with changing behavior over time, i.e. the actually provided service of a provider remained stable over time. Finally we did not account for the age of experiences, i.e. we assumed $age(i) = 0$ for all experiences i. Studying of those aspects is subject to our future research. The quality of the offer conformance prediction was measured in terms of its percental deviation from the actual offer conformance. Given a request r and an offer o the actual offer conformance was calculated as the mean offer conformance of all instances in o with respect to r. This is reasonable, since this is the expected offer conformance when executing the service belonging to o, presuming that all service instances are executed with equal probability. As already mentioned a provider offers a single service. He is characterized by the offer for that service and by the actually provided service. The latter is represented as an offer that differs from the advertised offer. The differences between the offer and the actually provided service are generated by replacing instances, uniformly chosen from the instances contained in the advertised offer, by others uniformly chosen from the basic instance set. We introduced a provider parameter indicating the maximum number of differing instances per offer. The plots are based on a provider where maximal 50% of the instances differ from the advertised offer. If executed, a service uniformly chooses one of its instances and provides it to the consumer. The consumer may rate this service by determining the match value of this instance with respect to the posed request. Afterwards the offer conformance can be calculated based on this rating and the match value for the whole service offer.

Algorithm 4. TestOfferConf($|E_{rel}|$,$minExpSim$)

1. create a request r and a set R of $|E_{rel}|$ requests having at least a similarity of $minExpSim$ to r
2. create a provider p based on a random offer o
3. create a set of $|C|$ comparison offers each having at least a match value of $minMV$ with the request r
4. **for** (each of the requests in R) **do**
5. generate an experience with p as well as the matching result vector based on the set of comparison offers
6. **end for**
7. get the offer conformance of the provider p with respect to r based on the generated experiences according to formula (2)
8. get the actual offer conformance of the provider p with respect to r as described in this section
9. **return** t he percental deviation of the predicted from the actual offer conformance value

The test was performed according to Alg. 4. It was run with several parameter settings and the average quality over 10000 runs was calculated. Figure 3(a) shows the quality of the offer conformance prediction depending on the minimal similarity between the given request and those the experiences are based on. The test was performed with the parameter values $|C| = 20$, $|I| = 100$ and $minMV = 0.5$. The quality is plotted for 20, 30 and 50 experiences. As expected the quality of the prediction increases with $minMV$, since the relevance of the considered experiences is higher. The optimal quality of about 10% deviation is reached for $minMV \approx 0.9$. When calculating the offer conformance by assigning the same weight to all experiences, the quality of the prediction is much worse. The dependency between the quality of the offer conformance prediction and



(a) (b)

Fig. 3. Quality of the offer conformance prediction depending on the minimal experience similarity (a) and the number of experiences (b)

the number of experiences is illustrated in Fig. 3(b). The test was performed with the parameter values $|C| = 50$, $|I| = 100$ and $minMV = 0.5$. The quality is plotted for similarity thresholds of 0.5, 0.7 and 0.9. As expected the quality of the prediction increases with the number of experiences and remains stable after a threshold specific number of experiences. This number is the smaller the smaller the minimal similarity of the experiences is. Considering an experience similarity of 0.9 we need about 50 experiences to have a prediction deviation of 10%. A number of 20 experiences is sufficient to have a prediction deviation of 12%. The described test series is based on the assumption that we know about the actual similarity of the requests, but in fact we have to rely on the approximate similarity when selecting the experiences to be considered for the offer conformance prediction. We can simply infer results for this case. Assuming a similarity deviation of 20% the average deviation of the offer conformance prediction is between 7% and 14% when having a minimal similarity of 0.8 and a number of 20 experiences (see Fig. 3(a)).

Based on a test series like this one may derive an appropriate function f defining the confidence of the prediction (see Def. 4). In our experiments requests and offers are created randomly. The distribution of their instances differs from that of real world service descriptions in that the match value distribution for the latter is smoother, i.e. the match values of the single instances within a request are more similar. Due to this observation we expect that the approximation of the request similarity and thus the offer conformance prediction will be even more accurate when working with real world service descriptions.

5 Conclusion

We introduced our approach for experience aware service ranking and selection. Designed as an extension it augments the functionality of existing matchmakers by allowing them to predict a service's future performance more accurately based on offer conformance experiences in former service interactions and thereby reducing the uncertainty encountered in this step. The services discovered by those matchers are then ranked based on their match values, the predicted offer conformance and the confidence of this prediction. The approach relies on subjective feedback in terms of ratings and considers the personalized nature of those experiences while avoiding explicit sharing of personal consumer information by comparing consumer preferences indirectly. It exploits available feedback effectively by considering not only ratings for a single service but also ratings for similar services of the same provider when evaluating the offer conformance of a specific service.

In our future work we plan to elaborate on mechanisms for providing request and preference templates as proposed in [3,15]. On the one hand this eases the creation of service descriptions, on the other hand it reduces the ambiguity of descriptions. Moreover it allows for more efficient storage and gathering of experiences. The presented approach considers experiences where service consumers rated a service as a whole. We plan to extend our solution by also allowing for refined experience, where consumers may rate partial aspects of a service. Another interesting point that is important when dealing with experiences of other consumers is that of dishonest experience providers. We have to question how to recognize those experiences, how to deal with them and how to prevent them. It is planned to adopt the reputation system solution of Obreiter et al. [16] for this purpose. Beside those extending features we will expand our evaluation by testing our solution with service descriptions based on existing description languages.

References

1. Küster, U., König-Ries, B.: Supporting Dynamics in Service Descriptions - The Key to Automatic Service Usage. In: 5th International Conference on Service Oriented Computing, Vienna (2007)
2. Josang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decis. Support Syst. 43(2), 618–644 (2007)
3. Kerrigan, M.: Web Service Selection Mechanisms in the Web Service Execution Environment (WSMX). In: 21st ACM Symposium on Applied Computing, Dijon, pp. 1664–1668 (2006)
4. Kokash, N., Birukou, A., D'Andrea, V.: Web Service Discovery Based on Past User Experience. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 95–107. Springer, Heidelberg (2007)
5. Billhardt, H., Hermoso, R., Ossowski, S., Centeno, R.: Trust-based Service Provider Selection in Open Environments. In: 22nd ACM Symposium on Applied Computing, Seoul, pp. 1375–1380 (2007)
6. Maximilien, E.M., Singh, M.P.: Conceptual Model of Web Service Reputation. SIGMOD Rec. 31(4), 36–41 (2002)
7. Sensoy, M., Pembe, F.C. (eds.): Experience-Based Service Provider Selection in Agent-Mediated E-Commerce. Eng. Appl. Artif. Intell. 20(3), 325–335 (2007)
8. Vu, L.-H., Hauswirth, M., Aberer, K.: QoS-based Service Selection and Ranking with Trust and Reputation Management. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3760, pp. 446–483. Springer, Heidelberg (2005)
9. Vu, L.-H., Porto, F., Hauswirth, M., Aberer, K.: An Extensible and Personalized Approach to QoS-enabled Service Discovery. In: 11th Intl. Database Engineering & Applications Symposium, Banff (2007)
10. Caballero, A., Bota, J.A., Gmez-Skarmeta, A.F.: On the Behaviour of the TRSIM Model for Trust and Reputation. In: 5th German Conf. on Multiagent System Technologies, Leipzig, pp. 182–193 (2007)
11. Manikrao, U.S., Prabhakar, T.V.: Dynamic Selection of Web Services with Recommendation System. In: Intl. Conf. on Next Generation Web Services Practices, p. 117. IEEE Computer Society, Washington (2005)
12. Wang, H.-C., Lee, C.-S., Ho, T.-H.: Combining Subjective and Objective QoS Factors for Personalized Web Service Selection. Expert Syst. Appl. 32(2), 571–584 (2007)
13. Küster, U., König-Ries, B.: Semantic Service Discovery with DIANE Service Descriptions. In: Intl. Workshop on Service Composition & SWS Challenge at the WI 2007, Silicon Valley (2007)
14. Klein, M., König-Ries, B., Müssig, M.: What is Needed for Semantic Service Descriptions? A Proposal for Suitable Language Constructs. Int. J. of Web and Grid Services 1(3/4), 328–364 (2005)
15. Stollberg, M., Hepp, M., Hoffmann, J.: A Caching Mechanism for Semantic Web Service Discovery. In: ISWC/ASWC, pp. 480–493 (2007)
16. Obreiter, P., König-Ries, B.: A New View on Normativeness in Distributed Reputation Systems - Beyond Behavioral Beliefs. In: 4th Intl. Workshop on Agents and Peer-to-Peer Computing, Utrecht (2005)