Enabling Trust-Aware Semantic Web Service Selection - A Flexible and Personalized Approach

Friederike Klan
Institute of Computer Science
Friedrich-Schiller-University of Jena
friederike.klan@uni-jena.de

Birgitta König-Ries
Institute of Computer Science
Friedrich-Schiller-University of Jena
birgitta.koenig-ries@uni-jena.de

ABSTRACT

In today's online markets, consumers need support in finding providers that offer the products or services they need and that are trustworthy. While Semantic Web Services (SWS) research addresses the first problem (discovering functionally suitable service providers), it neglects the second. Hence, several attempts have been made to complement service retrieval techniques based on semantic matchmaking with trust-establishing techniques that leverage collaborative consumer feedback. However, the diversity and multi-faceted nature of SWS impose special requirements on the underlying feedback mechanism, in particular w.r.t. their flexibility and expressiveness. Existing approaches only partially meet those requirements. In this paper, we will therefore propose a trust-establishing mechanism for Semantic Web Services that allows to assess a service provider's trustworthiness with respect to various service aspects and is flexible enough to adjust to various kinds of services and consumer requirements.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage And Retrieval**]: On-line Information Services

General Terms

Algorithms, Human Factors, Measurement

Keywords

trust-aware service selection, multi-criteria judgments

1. INTRODUCTION

Today, the WWW is a large market for products and services. Where the online market was pioneered by large companies, now Internet marketplaces such as eBay.com or Amazon.com allow anyone to easily set up a shop. This new business model offers rich opportunities for both sellers and consumers, but also raises new challenges, in particular for consumers. The huge amount and heterogeneity of information, products and services available online makes it difficult and sometimes even impossible for them to identify offers of interest to them. Hence, new techniques that support the user in the product search and selection process are required. In the past decade, semantic technologies have been developed and leveraged to approach this issue [7]. They provide information with a well-defined and machine-comprehensible meaning and thus enable computers to support people in identifying relevant content. This idea is not restricted to information, but also applies to functionality provided via the web as services. Semantic Web Services provide a specific functionality semantically described in a machine-processable way over a well-defined interface. Similarly, service requesters may semantically express their service requirements. Having both, a semantic description of a consumer's needs as well as the published semantic descriptions of available Web Services, suitable service offers can be automatically discovered by comparing (matching) the given service request with available offer descriptions. Services may be automatically configured and composed and finally invoked over the web.

Existing semantic matchmaking and service selection approaches evaluate the suitability of available service offers exclusively by comparing the published offer descriptions with a given request description. They implicitly assume that offer descriptions describe a service's capabilities correctly. This might have been a valid assumption in a market with a small number of well-known and accredited companies. However, it is no longer true in today's market, where easy and cheap access to the Internet and the emergence of online marketplaces that offer easy to set up online storefronts enable virtually everyone to provide his own online shop accessible to millions of buyers. As a consequence, service selection decisions that are purely based on the properties promised in offer descriptions are associated with uncertainty about the actual outcome of a service invocation. Trust-establishing techniques based on collaborative feedback [8] could be applied to infer missing knowledge about actual service outcomes and thus can reduce this uncertainty. Recently, several attempts have been made to complement service retrieval techniques based on semantic matchmaking with those techniques. However, the diversity and multi-faceted nature of SWS impose special requirements on the underlying feedback mechanism (Sect. 2). In particular, a viable solution should allow to judge a service

iiWAS2010 , 8-10 November, 2010, Paris, France

provider's trustworthiness with respect to various service aspects and should be flexible enough to adjust to different kinds of services and consumer requirements. Existing approaches only partially meet those requirements (Sect. 3). In this paper, we will therefore propose a flexible and personalized solution that enables trust-aware SWS selection by leveraging detailed consumer feedback (Sect. 5 and 6). Our solution is based on the semantic service description language DSD [11] and its matchmaking mechanisms (Sect. 4). As we will show, it effectively exploits available feedback to assess the trustworthiness of service providers and the SWS offered by them (Sect. 7).

2. REQUIREMENTS

Trust-establishing techniques based on collaborative consumer feedback seem to be very promising for enabling trust-aware Semantic Web Service selection. However, the diversity and multi-faceted nature of SWS impose special requirements on the underlying feedback mechanism. In the following, we will specify those requirements.

Consumer feedback is *subjective*, since it reflects a service provider's behavior as observed through a certain consumer's eyes. Hence, feedback is biased by personal expectations and preferences about the invoked service. Moreover, feedback may refer to different services of a provider and to different request contexts. For instance, a ticket booking service might have been used to buy group tickets for a school class or to buy a single ticket. However, the actual performance of a service might differ depending on the request context and hence the resulting feedback also does. Feedback mechanisms or SWS should account for those facts. To enable effective usage, feedback has to be *meaningful*, i.e., the expectations and the context underlying an expression of feedback should be clear. In addition, it should be evident whether and how feedback made under one circumstance can be used to infer about a service provider's behavior in another situation or even when providing another service.

We would also like to emphasize the necessity of feedback to be as *detailed* as possible, i.e. comprising of judgments referring to various aspects of a service interaction. The reasons are threefold. Firstly, feedback, judging the quality of a provided service as a whole, is of limited significance, since as an aggregated judgment it provides not more than a rough estimate of a service's performance. Secondly, trust requirements might differ among users. For instance, while one consumer cares much about a service provider's honesty with respect to the delivery time of a service, for another one the service's quality might be more important. Aggregated judgments do not allow to combine attribute-specific trust-information in a user-defined way and thus cannot adjust to differing trust requirements. Finally, aggregated feedback tends to be inaccurate. This is due to the fact, that humans are bad at integrating information about different aspects, as they appear in a multi-faceted service interaction, in particular if those aspects are diverse and incomparable [6]. In the context of detailed, i.e. multi-criteria, consumer feedback, meaningful also means that the relationship between different service aspects that might have been judged is clear and that all relevant aspects characterizing a certain service interaction have been judged. The latter is due to the fact, that inferred judgments based on incomplete information might be incorrect.

Another problem we encounter is feedback *scarcity*. Given certain service requirements, a certain context and a particular service, feedback for exactly this set-up is rare and typically not available at all. Hence, scarce feedback has to be exploited effectively. In particular, service experiences related to different, but similar contexts and those related to other services of a considered provider have to be leveraged.

When it comes to judge a service interaction, the willingness and ability to provide judgments might differ among consumers. Moreover, the type of service interactions to be judged and the trust requirements of users that use consumer feedback are diverse and not known in advance. Hence, a hard-wired solution, where the service aspects that have to be judged and for which subsequently trust information will be available are predefined, is inappropriate. In fact, the process of feedback elicitation should be *customizable* and should be *automatically configurable* at runtime.

3. RELATED APPROACHES

Evaluating a service's performance based on experiences that consumers made in former service interactions has a long tradition in trust and reputation systems and recommender systems research [8]. Recently, we observe a growing interest to apply those techniques to complement retrieval techniques based on semantic matchmaking. Typically, it is assumed that a list of services functionally matching a certain request is already available. The outcome of a feedback-based trust evaluation of these services' providers is then used as an (additional) criterion for ranking the retrieved services. Existing solutions in this line of research can be roughly classified into two groups that differ in the goal of feedback evaluation. The first group of approaches, e.g. [3, 9, 10] employs feedback-based techniques to compensate the shortcomings of purely semantic matchmaking techniques that result from the fact that request descriptions might be incomplete and inaccurate. For that purpose, consumer feedback is elicited during the (manual) service selection phase and is used later on to improve service retrieval. Feedback is either directly elicited by allowing consumers to judge the results provided by the semantic matchmaker or indirectly by learning from a consumer's service selection decisions in the past. However, since feedback is gathered before the actual service invocation, those approaches are not suitable to assess the trustworthiness of service providers. Moreover, eliciting direct consumer feedback before the actual service invocation presumes that users are able to decide about the suitability of a service being only provided with its formal offer description, which is unlikely. A second group of approaches relies on consumer feedback that is elicited after the service invocation and hence captures both, matching inaccuracies due to inaccurate and incomplete request descriptions and those resulting from inaccurate offer descriptions.

A considerable share of work [5, 13, 16, 17] investigated feedback-based techniques to evaluate a service's non- functional, i.e. Quality of Service (QoS), properties such as throughput or availability. QoS aspects form a subset of service attributes that in general is automatically measurable and where commonly agreed upon measuring methods exist. Hence, those approaches usually assume that consumer feedback is objective, i.e. not consumer-specific. However, as already argued, this assumption is no longer valid in scenarios that involve explicit user-provided feedback. Hence mechanisms that cope with the subjectivity inherent to this

type of feedback are required. Only a few approaches consider this aspect, e.g. [18] proposed to employ collaborative filtering techniques for that purpose. Those techniques allow to indirectly compare two consumers' tastes by comparing the judgments they have provided for the same services. Consumer feedback is also context-dependent. Several techniques to compare judgment contexts have been devised. For example, [12] suggests to employ collaborative filtering techniques for that purpose. In addition, many authors proposed to directly compare the service requests that lead to a judgment. Both, syntactic [10] as well as semantic request similarity measures [3–5, 16] have been proposed. However, none of the approaches takes consumer preferences into account.

The majority of existing trust-establishing solutions for SWS relies on single ratings judging the outcome of a service as a whole and thus do only partially meet our requirements. However, approaches that consider detailed consumer feedback have been also suggested, e.g. [13, 16, 17]. Typically, the set of service attributes that might be judged by consumers is either fixed for all services and hence rather generic or not appropriate for all service types, e.g. [16], or can be freely chosen from an ontology, e.g. [13, 17]. Ontological knowledge is used to integrate judgments referring to different service attributes. However, trust information are aggregated in a predefined way that does not account for individual trust requirements. A drawback of the mentioned approaches is, that they are not based on rich semantic descriptions, that are capable of describing the effect of a service. Instead, service requirements are typically described by means of desirable values for a set of attributes. As a consequence, those solutions are not able to capture the context of an attribute judgment, e.g. whether the execution time was judged for a weather service or for a service that offers the latest stock prices. Another issue is that the existing solutions typically require a user to judge all attributes of a service, i.e. do not account for a consumer's willingness and ability to provide judgments.

4. SEMANTIC WEB SERVICE RETRIEVAL

As a basis for further discussion, we introduce the semantic service description language DSD (DIANE Service Description) [11] and its mechanisms for automatic semantic service matchmaking that underlie our approach. Though DSD might not be as expressive as prominent logic-based semantic service description models such as OWL-S [1] or WSMO [2], its light-weighted approach turned out to be sufficiently expressive and well-suited for many practical application scenarios [3]. In contrast to logic-based approaches, DSD also provides an intuitive representation for service request and offer descriptions, that is comprehensible for the average user. Moreover, it provides language elements that allow to specify preferences related to the different service aspects and their importance.

Similarly to other service description approaches, DSD is ontology-based and describes the functionality a service provides as well as the functionality required by a service consumer by means of the precondition(s) and the set of possible effect(s) of a service execution. For better understanding, we illustrate this aspect using an example request from the well-known domain of product selling services (Fig. 1). In this service request, the desired effect is that a product is owned after service execution. A single effect corresponds to a particular service instance that can be executed. While service offer descriptions describe the individual service instances that are offered by a service provider, e.g. the set of mobile phones offered by a phone seller, service request descriptions declaratively characterize the set of service instances that is acceptable for a consumer. In the service request in Fig. 1, acceptable instances are mobile phones that are cheaper than 50$, are either silver or black, are of bar or slider style and are from either Nokia or Sony Ericsson. As

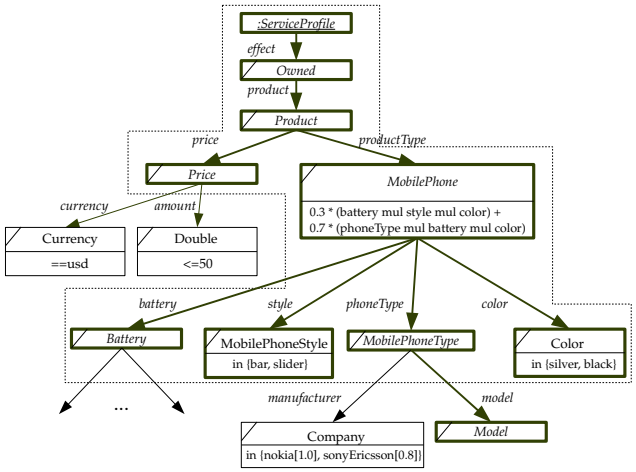

Figure 1: DSD service request

can be seen in the example, DSD utilizes a specific mechanism to declaratively and hierarchically characterize (acceptable) sets of service effects: Service effects are described by means of their attributes, such as price or color. Each attribute may be constrained by direct conditions on its values and by conditions on its subattributes. For instance, the attribute phoneType is constrained by a direct condition on its subattribute manufacturer, which indicates that only mobile phones from Nokia or Sony Ericsson are acceptable. Attribute conditions induce a tree-like and more and more fine-grained characterization of acceptable service effects. A DSD request does not only specify which service effects are acceptable, but also indicates to which degree they are acceptable. In this context, a preference value from $[0, 1]$ is specified for each attribute value. The default is 1.0 (totally acceptable), but alternative values might be specified in the direct conditions of each attribute. For example, the preference value for the attribute manufacturer is 1.0 for Nokia phones and 0.8 for mobile phones from Sony Ericsson.

The comparison of request and offer descriptions is iterative. Starting from the effect attribute of the request, the matchmaker checks in each step, whether the service effects described in the offer fulfill the conditions in the request. To illustrate the procedure, imagine that the request depicted in Fig. 1 is compared to a given offer. The matchmaker would first check whether the effects described in the offer match the type Owned as indicated in the request. If this is true, the attribute condition product is checked. Iterative proceeding to the leaves of the request, results in a

[1] http://www.w3.org/Submission/OWL-S/
[2] http://www.wsmo.org/TR/d2/v1.3
[3] http://sws-challenge.org

mismatch or match for each offered service instance and, in case of a match, in preference values for each attribute. In a final pass from the leaves to the effect root, those preference values are aggregated using aggregation functions specified for all intermediate request nodes. By default, the preference values for the subattributes are multiplied. However, alternative aggregation functions might be specified (see attribute productType in Fig. 1). The comparison algorithm outputs an aggregated overall preference value from $[0, 1]$, also called *matching value*, for each matching service instance. Based on their matching values, fitting service offers might be ranked and presented to the service consumer for selection.

5. ELICITING CONSUMER FEEDBACK

In the following, we will propose a collaborative feedback mechanism that allows to predict a SWS provider's trustworthiness. The main idea is that service consumers judge the quality of a service interaction with a certain provider by providing ratings for several service aspects. Those ratings are then used to infer knowledge about the provider's future behavior with respect to those aspects. This information might be used as an supplementary criterion when ranking semantically matching services or might be simply presented to the user in addition to the service results ranked by their matching values as determined by the semantic matchmaker. Note, that service consumers might have very different demands on the trustworthiness of a service provider. While one consumer might care much about the timeliness of a pizza delivery service, to another one the quality of the provided pizza might be more important. Having detailed, i.e. aspect-specific, information about the trustworthiness of a service provider, we can aggregate trust-information in a user-defined way.

In the following, we will first analyze what is required to make detailed consumer feedback meaningful, comprehensive and appropriate to characterize a certain service interaction. We will then demonstrate how semantic service descriptions can be used to elicit feedback that fulfills those requirements while still being flexible in the choice of what and how much service aspects to judge. A detailed discussion on how to support service consumers in the feedback elicitation process is out of the scope of this paper and is going to be published elsewhere. The focus of Sect. 6 is on how to effectively use elicited consumer feedback to assess a service provider's trustworthiness and thereby accounting for the subjectivity of judgments and their context-dependent nature.

What is required to make consumer feedback appropriate, comprehensive and meaningful?

We assume, that a service request at least covers all service aspects that are important to the consumer. Potentially, all service aspects in a request description are appropriate judgment targets and might be rated by a consumer. However, in order to be able to exploit these ratings, we need to make sure that they are meaningful and comprehensive. In addition, we need to know how hierarchical aspects of ratings relate to each other. We illustrate those issues with an example. Suppose a provider p offers a high-quality cell phone. Imagine that we have consumer feedback referring to p consisting of a single quality rating, which indicates that the quality was as promised. Unfortunately, this does not allow us to infer that p will provide the offered cell phone in the promised quality, since we do not know enough about the request context, e.g. to which product the quality rating refers to. Consequently, to obtain meaningful feedback, we need to store the context of the provided judgments. Assume that the cell phone service is also described in terms of its price. In this case, we need ratings for both price and quality to ensure correct interpretation of ratings. A consumer who got the promised phone at a higher than advertised price would rate the quality as good and the price as bad. We need to ensure that we don't lose that information. Thus, the set of ratings needs to be comprehensive, i.e., cover all aspects that are important to the requester. Now, suppose that a consumer rated usability, material and battery capacity of an offered cell phone. To leverage the provided ratings for a request asking for a high-quality phone, we need to know how those aspects relate to each other. The challenging question is how to fulfill the identified requirements while still being flexible in the choice of the aspects to rate.

Creating appropriate, comprehensive and meaningful consumer feedback

We propose the concept of a *feedback structure* to deal with that issue. A feedback structure is a subtree of the request tree, whose leaves correspond to the aspects that may be rated by the user. Consider the example request depicted in Fig. 1. The dotted part of the tree indicates a possible feedback structure for that request, where the aspects price, battery, style, color and phoneType have to be rated by the consumer. Note that this structure contains all information that are necessary to effectively utilize the provided ratings. In particular, it encodes the context of a rating in terms of the path from the request root to the rated aspect, the other aspects that were judged and the hierarchical relationship between the considered aspects.

To assure that the provided feedback is comprehensive, the request subtrees rooted at the feedback structure's leaves should cover all leaves of the tree. This guarantees that all service aspects considered in the request description are either directly or indirectly (by providing an aggregated rating) judged by the service consumer. The feedback structure depicted in Fig. 1 fulfills this requirement and thus is valid. Omitting, e.g., the aspect phoneType would result in an invalid structure. Note, that we are still flexible in the choice of the attributes to be rated, e.g. we could allow the consumer to provide a single rating for productType instead of asking him to judge battery, style, color and phoneType separately. Note, that providing a single overall service judgment for a service interaction is also possible using this scheme. In this case, the feedback structure comprises only of the request's root node. Obviously, there is a trade-off between the detailedness of the provided judgments and the rating effort. The proposed approach enables consumers to make this compromise according to their personal judgment preferences.

Once a consumer provided ratings for all service aspects corresponding to the leaves of the feedback structure, we use the aggregation functions specified in the service request to determine aggregated ratings for all internal nodes of the feedback structure. The feedback structure together with the ratings of the consumer are propagated to other consumers and might be used to infer knowledge about the provider's future behavior.

6. UTILIZING CONSUMER FEEDBACK

How to utilize consumer feedback based on various feedback structures to infer knowledge about a service's future performance? Imagine, we have posed request r depicted in Fig. 1 and wish to know what service quality we can expect when using service s offered by provider p. More formally, assuming we had already executed service s, how would we have judged its quality? We propose to leverage feedback provided by consumers that interacted with service provider p in the past to predict this judgment. However, as argued before, feedback items are of different value for us. Judgments that refer to a service that is similar to s and that are made by consumers that had similar requirements and thus rate a given service similarly are more valuable and hence more relevant for us.

How to determine relevance?

We propose to apply techniques from user-based CF [15] to approach this issue. Those systems aim at predicting a consumer's unknown rating for a product by utilizing ratings of other users that have a similar taste. Those neighboring users are identified by comparing the ratings they provided for different purchased items with those of the target user. The underlying assumption is that the more similar the ratings are, the more similar are the users' tastes and the more similar their ratings for other products will be. How can we apply this principle to solve our problem? If we want to know whether the judgments of a feedback provider c for services of provider p would be similar to ours when judging the same service interaction, we need to compare the ratings c would give for different services of p with ours. However, we do not have those ratings at our disposal. To solve this problem, we make use of the fact, that we have explicit models of both, c's service requirements, namely his service request, and the service offers of p. Hence, we can employ the matchmaker to determine those judgments. More specifically, to determine how similar the requirements of c are to ours, we compare the matching values of p's services with respect to c's service request with those of p's services with respect to our service request. A similar technique, item-based CF [15], could be applied to determine the similarity of two services. For that purpose, we have to compare the matching values provided for the two services with respect to different requests. The more similar those values, the more similar are the services. Note that the proposed procedure does not require additional computational effort since the required match values are determined during the service selection process anyway.

How are provider and service similarity determined?

Suppose we have posed the request r and wish to assess the performance of service s offered by provider p. Assume, that we have a set F of feedback items provided by other consumers to perform this task. In a first step, we create a feedback matrix *FM* with a row for each available feedback item $f \in F$ and a column for each service s' offered by provider p. Suppose that the feedback item f was based on the request r'. Then the entry *FM(f,s')* contains the matching value of r' and s'. To determine how similar the rating provided in f is to the one we would have given having used the same service, we have to compare the row *FM(f)* with the matching results we obtain for our request r. Similarly, to determine how similar the rated service s' is to the service s, we are interested in, we have to compare the columns for s and s'. We implemented our approach with one of the most prominent, namely the correlation-based similarity measure proposed in [14]. Once we have determined the relevance of each feedback item referring to the considered provider p, we are ready to predict a rating for service s using those information.

```
1  applyFeedback(Request r, Service s) {
2     – get all feedback F that is available for the provider p of s
3     – create the feedback matrix FM := (a_ij) with i in F
4        and j in services(p)
5     for each (item f in F) {
6        – determine the similarity of s and the service
7          judged in f
8        – determine the similarity of the request underlying
9          f and r
10    }
11    – if possible, predict a rating for each attribute in r
12 }
```

Listing 1: Using feedback to predict a rating for the service s of provider p in the context of a given request r

Feedback integration

Since feedback might be based on various feedback structures, the challenging question is how to integrate this feedback to infer knowledge about a service's future performance? Consider again, that we have posed the request r depicted in Fig. 1 and wish to predict attribute ratings for service s offered by provider p. Suppose we are provided with a feedback item f based on the feedback structure *fs* depicted in Fig. 1 (highlighted part) provided by consumer c. Now, imagine, we would like to know how s performs with respect to the aspect product. To answer this question, we have to check whether the feedback structure *fs* contains the aspect product within the same context, i.e., having the same path as the aspect product in r. Fortunately, this is true. However, consumer c did not directly rate the product attribute. Hence, we have to use the aggregated ratings stored in *fs*. Note, that those ratings have been aggregated using the aggregation functions specified in c's request. Those functions imply a weighting of aspects, which is specific for c and reflects his personal preferences. Hence, aggregated ratings are a source of subjectivity in consumer feedback. To mitigate that problem, we suggest to use direct consumer ratings if possible and aggregate them using the aggregation function defined in our own request. With respect to our example, this means that, if possible, we should have utilized consumer c's (direct) ratings for price, battery, style, phoneType and color to aggregate them to a product rating using the corresponding aggregation function defined within our request.

However, as already argued, it is advisable to consider ratings that judge all aspects that are important in the context of our request to not miss a potentially important aspect of a service interaction. We ensure this property by demanding, that the request subtrees rooted at the attributes used for the rating prediction cover all leaves of our request tree. We call the maximal subtree of the request that exhibits this property a feedback cover, where maximal refers to the number of request nodes it contains. Considering again our example, the feedback cover is the dotted part of the request.

Note, that it does not contain the model node from *fs*, since *c* did not rate manufacturer, which is also important for us. Once having determined the feedback cover for the feedback item and the given request, we extract ratings for all feedback cover leaf nodes from the feedback item and determine aggregated ratings for all intermediate feedback cover nodes using the aggregation functions specified in our request. We repeat this procedure for all feedback items available for the considered service provider p. Ultimately, we end up with a set of ratings for each request node stemming from the available feedback items. Note, that for aspects which are not contained in any of the feedback covers, no ratings are available and thus no rating prediction can be made. In a final step, the unkown attribute ratings for the given request r and service s are calculated as an aggregate of the ratings contributed by the single feedback items $f \in F$. As usual in CF [1], the value of the rating $rat(a)$ for a certain service aspect a is determined as

$$rat(a) = \overline{rat(a)} + k \sum_{f \in F} w(f) \cdot (rat_f(a) - \overline{rat(a)}), \qquad (1)$$

where $\overline{rat(a)} = \frac{\sum_{f \in F} rat_f(a)}{|F|}$ and $k = \frac{1}{\sum_{f \in F} w(f)}$ is a normalizing factor. The weight $w(f)$ of a certain feedback item $f \in F$ indicates its relevance for the prediction and is determined as the product of feedback provider and service similarity $sim(f)$ of this feedback item. The listings 1 and 2 summarize the introduced procedures for feedback usage and rating prediction.

```
1  predictRating(Request r, SetOfFeedback F, Similarities S_F)
2     for each (item f in F)
3        – determine the feedback cover C
4        – get the ratings for the cover leaves
5        – determine the aggregated ratings for all other nodes in
6          C using the aggregation functions defined in r
7     }
8     for each (attribute a in r) {
9        – predict a rating using the single ratings for a and
10         the similarities S_F(a)
11    }
```

Listing 2: Rating prediction for a request r using the feedback items in F and their feedback provider and servcie similarities S_F

Improved similarity calculation

So far, we used the same weight for each rating contributed by a feedback item f. However, this strategy does not account for the fact that judgments for attributes that are very similar to the corresponding request and service attribute are more valuable for the rating prediction and thus should be weighted higher. As a solution, we propose to weight each attribute rating of a feedback item individually. In this context, the weight for a rating that judges aspect a is determined as the product of the contributing feedback item's overall similarity $sim(f)$ and the feedback item's similarity with respect to aspect a. Attribute similarity values are determined by maintaining an individual feedback matrix for each request attribute. Those feedback matrices are composed of the matching results for the corresponding subattributes of a request. The proposed weighting scheme ensures that an attribute is only weighted high, if both, the similarity of the feedback item as a whole and the individual similarity of the rated aspect are high. In contrast to our original solution, we compute the overall similarity $sim(f)$ of a certain feedback item f as the mean over the single attribute similarities contained in the feedback cover determined by f. As indicated in [2], this can significantly improve the prediction quality. Our evaluation shows that leveraging individual similarity values for each attribute rating results in significant improvements of the prediction quality.

7. EXPERIMENTAL RESULTS

We evaluated our approach using information about services selling computer items. However, since real user data, in particular real service requests as well as information about the cheating behavior of service providers are not available, we had to appropriately model those aspects to generate the required test data.

Test data

For the purpose of the evaluation, we extracted information about around 7000 computer items (from 8 categories) and their attributes from the WWW and created DSD service descriptions for them. Those items where randomly assigned to service providers, where each provider offered between $5 - 10$ items of each category. The items where uniformly chosen from the available items in each category. To simulate misbehavior, we generated modified versions of each provider's items. This was done by changing an item's attribute values with probability 0.8 by an amount differing between 0 and 50% of the corresponding attribute value of the original item. While the descriptions of the original items (representing a provider's promised service) where used to determine an offer's matching degree with a given request, the actual output of the service was determined by the modified pendants of the invoked service instance. More specifically, we interpreted the matching value of the modified service instance as consumer rating. We generated service requests covering typical requirements of consumers looking for computer items. Consumer preferences over single attributes were modeled using a decreasing/increasing (depending on the attribute type) linear function over an attribute's possible values. For instance, in case of the price attribute, the preference value linearly decreased from 1 for the lowest possible price to 0 for the maximal acceptable price and remained 0 for higher prices. The maximal/minimal acceptable attribute value was randomly chosen from the range spanned by existing item attribute values. Individual attribute preferences where aggregated using a weighted mean, where the weights where uniformly chosen from $[0, 1]$. We would like to mention, that we are aware of the fact that this kind of test data are artificial to some degree. However, we argue that the service providers and requests in the generated test set or much more diverse than those one would expect in reality and thus make it harder rather than easier for our system.

Test settings

To generate consumer feedback, 100 requests and 20 providers were created. For each request, we determined all matching service instances among those offered by the generated providers and uniformly chose one of them for invocation. We stored the matching results for all instances offered by

the provider of this instance (determined using the descriptions of the original items) and noted the actual performance of the provider given by the matching results of the selected instance's modified version. For each of the 100 pairs of generated request and invoked service instance, we used our proposed procedure to predict the performance of the corresponding service provider with respect to the given instance and request, thereby using the generated consumer feedback. In average, 5 feedback items were used for each prediction. To assess the quality of the prediction, we compared the predicted overall rating for the invoked service instance with the matching value of the corresponding modified service instance and determined the average absolute deviation of the predicted from the actual matching value (prediction error). The results were averaged over all 100 pairs.

We ran tests with different variations of our prediction algorithm and different types of feedback. The tests *SFb0* and *SFb* served as a baseline for our evaluation. Both tests used only overall service judgments for the rating prediction, i.e. did not consider any attribute ratings. While test *SFb0* did not use any similarity information for the prediction (the single ratings were simply averaged), test *SFb* employed overall service and provider similarity to weight different judgments. We also ran two tests that used detailed consumer feedback. The tests differed in the type of similarity information that was used for the prediction. While test *DFbOvSim* performed our basic prediction algorithm, i.e. considered only the overall service and provider similarity to weight different feedback items, test *DFbCombSim* used an implementation of our improved prediction algorithm, that leveraged similarity information about the single service attributes.

Results

Test variation *SFb0*, that did not use any similarity information, performed significantly worse than all other test runs (Fig. 4). Hence, the quality of the similarity information has a large impact on the prediction quality supporting our hypothesis that this is the case.

In a first series of tests, we wanted to find out whether both, feedback provider and service similarity, have an impact on the quality of the rating prediction. We ran tests using different similarity thresholds. Fig. 2/3 shows the absolute prediction error for tests with a service/provider similarity threshold of 0.0 and increasing thresholds for the feedback provider/service similarity. As can be seen, while the prediction quality increases with both, increasing threshold for feedback provider and service similarity, the effect of the feedback provider similarity was much higher. However, using high thresholds for both similarities achieved an even higher reduction of the prediction error (Fig. 4). Hence, we conclude that both effects enhance each other.

In a second series of tests, we investigated whether detailed consumer feedback improves the prediction quality and if so how well the different approaches to similarity calculation performed. Fig. 4 (left) shows the results for runs that leveraged all available feedback for the rating prediction, while Fig. 4 (right) depicts the results for runs that used only feedback with a high value (0.8) for service and feedback provider similarity. The results for the tests *SFb* and *DFbOvSim* indicate, that the prediction quality only minimally improves when detailed consumer feedback instead of a single overall rating is used for the rating prediction. Note, that this improvement is solely attributed to the fact that the predicted overall rating was determined by aggregating the ratings for the single attributes in a request specific way. At this point, similarity information about the single attributes that were rated has not been used. To evaluate the effect of improved similarity calculation on the quality of the prediction, we performed test *DFbCombSim*. As can be seen in Fig.4, leveraging similarity information about the single attributes significantly improved the prediction quality. It reduced the prediction error produced by the tests *SFb* and *DFbOvSim*, to about 1/5. All test variations achieved a high reduction of the prediction error when leveraging only feedback with a high similarity (Fig. 4 right). However, using similarity information about single attributes allowed to exploit available feedback more effectively. This thesis is supported by the results depicted in Fig. 4 (left). While all test variations used the same feedback information, the prediction error was much smaller when similarity information about single attributes were considered. However, the performance of the rating prediction re-

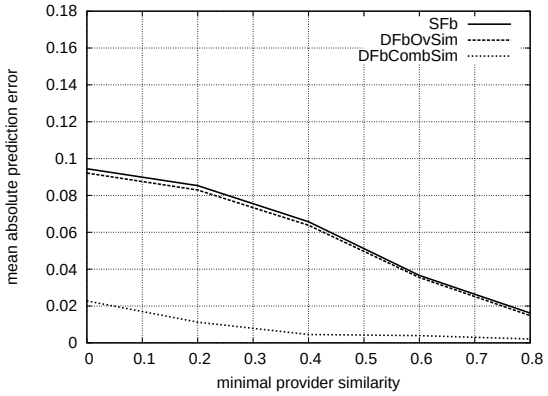

Figure 2: Prediction error with increasing thresholds for feedback provider similarity

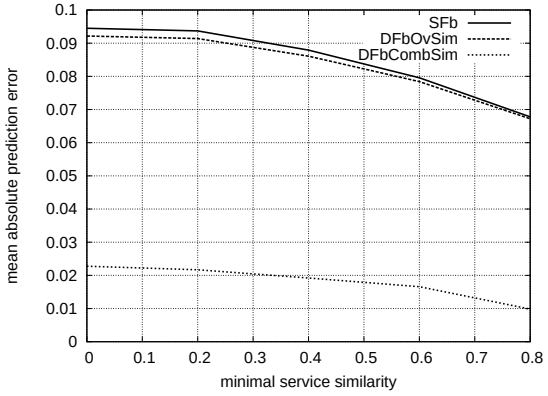

Figure 3: Prediction error with increasing thresholds for service similarity

lies on the availability of sufficiently detailed consumer feedback. The more detailed the provided feedback, the higher the prediction quality. Studying the relationship between the detailedness of consumer feedback and the prediction quality is subject to our future work.

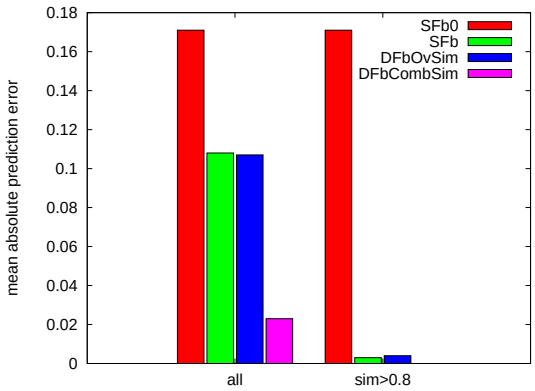

Figure 4: Prediction errors for the 4 test variations

8. CONCLUSION

We have presented an approach that combines techniques from SWS with methods known from reputation systems and CF to elicit meaningful feedback from service consumers that can then be used to support other consumers in their service selection. With this approach, providers can be selected based not only on the functionality of the services they offer, but also on their trustworthiness with *respect to a specific request and service*. The latter is a decisive advantage as trustworthiness in online markets is highly context-dependent. At the same time, our approach overcomes the problem of scarce feedback by flexibly finding relevant feedback to a given request. The evaluation shows that this results in very low error rates and thus offers considerable support for users in their decision making. The main advantage of our solution is its flexibility. It is *applicable to arbitrary types of services* and dynamically *adjusts to consumers with differing trust- and judgment requirements.*

References

[1] G. Adomavicius. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Trans. on Inf. Sys.*, 23(1):103, 2005.

[2] G. Adomavicius and Y. Kwon. New Recommendation Techniques for Multi-Criteria Rating Systems. *IEEE Intelligent Systems*, 22(3), 2007.

[3] A. Averbakh, D. Krause, and D. Skoutas. Exploiting User Feedback to Improve Semantic Web Service Discovery. In *8th International Semantic Web Conference (ISWC 2009)*, 2009.

[4] H. Billhardt, R. Hermoso, S. Ossowski, and R. Centeno. Trust-Based Service Provider Selection in Open Environments. In *22nd ACM Symposium on Applied Computing*, pages 1375–1380, Seoul, 2007.

[5] A. Caballero, J. A. Botía, and A. F. Gómez-Skarmeta. On the Behaviour of the TRSIM Model for Trust and Reputation. In *5th German Conf. on Multiagent System Technologies, Leipzig*, pages 182–193, 2007.

[6] R. M. Dawes. The Robust Beauty of Improper Linear Models in Decision Making. *American Psychologist*, 34(7):571–582, 1979.

[7] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue. *Enabling Semantic Web Services: The Web Service Modeling Ontology.* Springer, 2007.

[8] A. Jøsang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decis. Support Syst.*, 43(2):618–644, 2007.

[9] M. Kerrigan. Web Service Selection Mechanisms in the Web Service Execution Environment (WSMX). In *21st ACM Symposium on Applied Computing*, pages 1664–1668, Dijon, 2006.

[10] N. Kokash, A. Birukou, and V. D'Andrea. Web Service Discovery Based on Past User Experience. In *BIS*, volume 4439, pages 95–107. Springer, 2007.

[11] U. Küster, B. König-Ries, M. Klein, and M. Stern. DIANE - A Matchmaking-Centered Framework for Automated Service Discovery, Composition, Binding and Invocation. In *WWW*, 2007.

[12] U. S. Manikrao and T. V. Prabhakar. Dynamic Selection of Web Services with Recommendation System. In *Intl. Conf. on Next Generation Web Services Practices*, page 117, Washington, DC, 2005. IEEE Computer Society.

[13] E. M. Maximilien and M. P. Singh. Agent-Based Trust Model Involving Multiple Qualities. In *AAMAS*, pages 519–526, 2005.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. pages 175–186. ACM, 1994.

[15] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative Filtering Recommender Systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, Lecture Notes in Computer Science, pages 291–324. Springer, 2007.

[16] M. Sensoy, F. Pembe, H. Zirtiloglu, P. Yolum, and A. Bener. Experience-Based Service Provider Selection in Agent-Mediated E-Commerce. *Eng. Appl. Artif. Intell.*, 20(3):325–335, 2007.

[17] L. H. Vu, F. Porto, M. Hauswirth, and K. Aberer. An Extensible and Personalized Approach to QoS-enabled Service Discovery. In *IDEAS*, Banff, 2007.

[18] H. C. Wang, C. S. Lee, and T. H. Ho. Combining Subjective and Objective QoS Factors for Personalized Web Service Selection. *Expert Syst. Appl.*, 32(2):571–584, 2007.