A Fuzzy Model for Reasoning about Reputation in Web Services

Wanita Sherchan, Seng W. Loke, Shonali Krishnaswamy
Faculty of Information Technology, Monash University
Caulfield East, Victoria 3145, Australia
{wanitas,swloke,shonali}@csse.monash.edu.au

ABSTRACT

Reputation systems are typically based on ratings given by the users. When there are no mechanisms in place to detect collusion and deception, combining user testimonies as to form a provider’s reputation may not give an accurate assessment, especially if the context of the ratings is not known. Moreover, such systems are vulnerable to manipulations by malicious users. Hence it becomes essential to establish the validity of the ratings prior to using them in formulating reputation based on such ratings. It is important to identify the rationale behind the ratings so that similar ratings (or ratings pertaining to a context) can be aggregated to obtain a reputation value meaningful in that context. We propose a fuzzy approach to analyze user rating behavior to infer the rationale for ratings in a web services environment. This inference of rationale facilitates the system to validate ratings, detect deception and collusion, identify user preferences and provide recommendations to users.

Categories and Subject Descriptors
I.2.3 [Artificial Intelligence]: Deduction and Theorem Proving—{inference engines, fuzzy reasoning}

Keywords
Rationale inference, reputation, web services

1. INTRODUCTION

Reputation systems introduce accountability in online interactions by creating an expectation that past history will be taken into account while making trust decisions [10]. Hence, it is expected that reputation systems will provide a fair, equitable and transparent evaluation of the reputation of the users/services. Furthermore, different users have different preferences and a service suitable for one user may not be suitable for another. Hence reputation systems need to be reliable, transparent and yet able to cater for user preferences. Effective reputation systems enforce users to behave in a responsible way, provide a representative view of the user’s reliability, discourage and penalize deceitful behaviour and are tailored to cater for user preferences [10].

Various models for evaluating, updating and maintaining reputation have been proposed in the literature. Most of these systems [6, 7, 11, 15, 17], take user ratings as the general basis for reputation calculation and use various mechanisms of aggregating the ratings to form the reputation of the provider. When the system has no mechanism of validating the ratings, any mechanism of combining user testimonies to form reputation fails to give a true evaluation of a provider’s trustworthiness, reliability and credibility. Furthermore, such systems are prone to attack by malicious entities which may give false ratings and are hence susceptible to collusion and deception among parties [1, 13].

In context of web services, it is possible to monitor the actual performance of a service. Prior to each interaction, the service provider and the service requestor agree on desirable performance level, quantifying and enlisting it in a formal document called Service-Level Agreement (SLA) [8]. The SLA enlists the level of service quality that the provider guarantees to deliver in a number of parameters such as availability, performance, throughput and response time and also the clauses explicating penalties in case the service fails to comply with the agreed levels. To enforce these penalties performance monitoring mechanisms [5] are in place. Monitoring of SLAs ensures that the provider/service conforms to the agreed SLA. If a deviation occurs, or the agreed level is not met, then the relevant compensatory clause can be activated so that the provider pays for the violation of the contract. Based on such measured performance [5], objective performance measures compliance (ability of a service/provider to deliver agreed levels of QoS) and verity (consistency in delivering the agreed quality levels) have been formalized and quantified in [4].

Therefore in web services domain, two dimensions of reputation exist- (i) the subjective dimension - the users have the opportunity to rate the services they use so that they can express their views directly about the services they experienced, and (ii) the objective dimension - the objective performance measures compliance and verity that quantify the compliance levels and their variance thereby reflecting the actual performance history.

Since reputation is inherently a subjective notion, user ratings are an important consideration in reputation evaluation. However, reputation evaluation based only on user ratings suffer from problems such as lack of incentive (for
users) for leaving feedback, (users’) general bias towards positive ratings and probability of unfair (unfairly positive or negative) ratings [3]. Since users do not have incentive to leave any feedback, many users refrain from doing so; this may affect reputation calculation, even making it biased. In addition, due to fear of retaliation or just by human nature to be “nice”, most ratings tend to be positively biased, unrelated to the actual performance [3]. Furthermore, there may exist problems such as collusion and deception [1, 13]. Therefore, it is reasonable that some effort is focused towards reconciliation of the two complementary aspects of reputation.

In this paper, we investigate the relationship between these two dimensions; using the objective performance measure to determine if the subjective view is rational (i.e. whether it matches the objective view). We use the objective measure compliance to verify and strengthen the subjective measure user rating. Combining these two dimensions of reputation we propose a fuzzy approach to inferring rationale for ratings. Based on such inferred rationale we are able to support detecting deception and collusion, identifying user preferences and providing recommendations to users.

The rest of the paper is organized as follows. Section 2 presents the need for rationale inference through illustrative scenarios. Section 3 describes our fuzzy approach proposed to establish rationale. Through simulated scenarios section 4 illustrates how our fuzzy model infers rationale. Section 5 explains how rationale inference can be incorporated in existing web services architecture. Finally, section 6 presents the conclusions and future work.

2. THE NEED TO ESTABLISH RATIONALE

Every rating given by a user has some rationale behind it. It may be difficult, even for the user, to explicitly state the rationale; nevertheless the rationale is implicit in the rating. When there is no mechanism of analyzing ratings, it becomes difficult to extract the rationale from the ratings. Extracting rationale can be very useful for different purposes such as validating ratings, identifying user preferences, detecting collusion and detecting deception. Currently there has been little effort focused towards rationale inference and reasoning in the context of reputation. We aim to bridge this gap to facilitate a better understanding and holistic treatment of reputation in the specific context of Web Services. With the following motivational scenarios we illustrate how rationale inference is useful and explain how our rationale inference approach helps eliminate existing problems.

2.1 Illustrative Scenario 1

Consider the hypothetical scenario where a user U1 gave a rating of 4 (where 1 is lowest and 5 is highest) to a service WS1. By merely superficial examination of the rating only, it is not known what context the rating refers to, or why the user gave that particular rating. The user may have given a high rating to the service simply because the service gives the lowest prices and price is the most important factor for that user. The user may be willing to compromise on a few other parameters but not on the price, but this fact may not be reflected on the SLA. When the actual interaction took place, the service may have had a long response time, but the user did not mind, and gave a high rating to the service, because it always gives the best price. Hence the rationale here is the user preference for a particular attribute (price). If there is any mechanism of identifying such rationale, then this rating is relevant only to those users who have the same preference. Existing reputation systems either do not consider preferences [11, 14, 15, 17] or ask users to explicitly state their preferences [6]. It is well established that explicit specification of preferences is considered difficult (particularly) when multiple parameters are considered [12]. Therefore it is more desirable to infer user preferences from their rating behavior [2]. Collaborative filtering recommender systems [2] follow this approach. Another approach is comparing the ratings given by the user with the actual performance of the service, where such information is available. In this paper we take the latter approach. Inferring user preferences facilitates in making suitable recommendations to users by aggregating ratings given by users with similar preferences. It is also useful in distinguishing major features of the services, for example, a service WS2 may have been rated highly by users for whom “accuracy” is the most important factor, then service WS2 can be recommended as service WS2 is good for users who want good accuracy.

2.2 Illustrative Scenario 2

Consider another scenario where a user gives high ratings to a service repeatedly although the statistics show that the service did not meet the agreed level in most of the parameters. The service performance was low in all of the parameters in multiple invocations of the service, but the user always gives high ratings. In this situation, the user could be colluding with the provider to boost its reputation, so in this case the rationale could be collusion. Therefore, if this is the case, then the ratings given by such a user should be identified as invalid and weeded out. This reasoning can be inferred only from the comparison of the ratings with the performance statistics. Most existing reputation models do not detect such situations and simply incorporate the ratings into reputation calculation, thereby enabling dishonest users to manipulate the system. There have been few proposals for filtering out unfair ratings. Yu and Singh [16] propose a mechanism for detecting and immunizing deception in a multiagent distributed reputation management scenario where an agent combines testimonies from several witnesses to determine its rating of another agent. Deception detection is based on the assumption that raters with low reputation are likely to be deceptive and hence their testimonies should be given lower weights. This approach assumes that raters lie consistently; which may not be the case in real life situation - agents may lie only in certain situations, such as when the payoff is high. Other works that deal with deception in a multi-agent distributed environment are by Dellarocas [1] and by Whitby et al [13]. These works are based on the assumption that deceptive ratings can be recognized by their statistical properties only [13], i.e. ratings that are deviating from the normal are unreliable; and hence use cluster filtering to reduce the effect of unfair ratings. However, in a Web Services context, service parameters are unlikely to be fixed or static (e.g. QoS conditions), hence, variability in user opinion may be a valid occurrence. Statistical methods of deception detection do not consider such occurrences and may incorrectly term these ratings as invalid. Thus these methods are not very suitable in Web Services domain. Moreover, the accuracy of these systems is affected by the proportion of unfair raters and high percentage of unfair raters may even make the sys-
tems counterproductive [13]. Hence there is a need for a mechanism for deception detection that is tailored for Web Services context and is independent of both the consistency in user’s rating behavior and the percentage of unfair raters so that the system (deception detection) performs well in all situations. We contend that analyzing ratings with respect to the actual performance of the service is such an approach and provides mechanism for deception detection as well as inferring and explicating rationale for ratings.

2.3 Significance of Rationale Inference

Analysis such as above (in scenarios 1 and 2) are feasible because it is possible to have access to the actual performance statistics [5] and the Service Level Agreement (SLA), and the deviation from the SLA can be calculated [4] from which estimation can be made about an appropriate rating of the service. The rating provided by the user can then be compared with this estimation to infer the rationale for the users ratings. Thus explicated rationale can then be used for detecting deception, validating ratings, detecting collusion, identifying user preferences and providing recommendations to users. In summary, rationale inference has following key implications:

- Rating validation can be used to filter out false ratings from reputation calculation to get an evaluation that is more indicative of the performance of the providers. Weeding out false ratings also helps in preventing deception and collusion among parties, which makes the system more robust from attacks by malicious entities.

- Based on the bias of the user, it is possible to tailor reputation calculations to the particular needs of the user, i.e. rating aggregation based on user bias. For example, a user requests for reputation evaluation of some providers. The system has inferred from the user’s rating behavior that s/he places high importance on response time. Then the reputation evaluation can be tailored to her/his bias by using only the ratings given by the users for whom response time is the most important factor. Similarly, suitable recommendations can be made based on the bias of the user. For example, if the system knows that a user (A) is biased towards response time, then a service which has been rated highly by other users with same bias can be recommended to the user (A).

3. A FUZZY APPROACH FOR INFERRING RATIONALE FOR USER RATINGS

3.1 A Fuzzy Approach

In this section we introduce our fuzzy logic based reasoning model for inferring rationale using ratings and past performance. As a simple performance measure we use the compliance of a service. Compliance refers to the ability of the service provider to deliver the level of service quality as agreed in the SLA and is calculated as the normalized difference in the delivered and projected values of each QoS attribute [4].

Let \( a \) be an attribute, \( a_p \) be projected value of \( a \) as agreed in the SLA and \( a_d \) be delivered value of \( a \) as obtained from the performance monitoring system. Then, compliance of the attribute \( a \) when the service is invoked the \( j^{th} \) time, represented as \( C_j \), is calculated as the normalized difference in the values of \( a_p \) and \( a_d \) [4]:

\[
C_j = \frac{(a_d - a_p)}{a_p}
\]  

The compliance value can be negative or positive and scales with different values of projected and delivered values. Ideally, compliance is zero which indicates that the provider is fully compliant with the agreed values. When compliance is positive, the provider is highly compliant as it delivered more than it promised. When compliance is negative, the provider is non compliant as it did not deliver what it guaranteed. Thus the relationship between compliance and rating is formulated as follows:

- If \( C_j < 0 \), then the service has low compliance hence it should have low rating
- If \( C_j = 0 \), service is compliant hence it should have high rating
- If \( C_j > 0 \), service is highly compliant hence it should have very high rating

This three-level categorization of Compliance is coarse grained (i.e. at a very general level of abstraction). It does not reveal the degree of compliance and may not differentiate between various compliance levels of different providers. For example, a compliance value of 0.0001 or less is very near the compliant or zero value of compliance, and hence should be treated as such. However, the categorization such as above does not consider this and may term these as highly compliant (for compliance = 0.0001) or non compliant (for compliance = -0.0001), which is unreasonable. Hence there is a need for a more fine grained measure of compliance.

In addition, we propose to estimate a rating for the service/provider based on the compliance values and compare this value with the rating given by the user. For both of these purposes, fuzzy representation provides a suitable solution. Furthermore, terms used to refer to both compliance (such as “compliant”, “very compliant”) and ratings (“poor”, “moderate”, “good”, “very good”) have a fuzzy element; hence fuzzy representation gives a more realistic mapping between the two.

Having established the rationale for using fuzzy inferencing in this problem, in the next section we describe the mapping of our problem into a fuzzy inferencing process [9].

3.2 Mapping to Fuzzy Inference Process

We propose to calculate the compliance values for each parameter specified in the SLA and combine them using fuzzy inference rules to get the overall estimated rating for the service in that invocation. We propose to estimate a set of ratings for a service based on its compliance in the different SLA parameters. Each rating is estimated on a certain basis, for example, unbiased towards any parameter, biased towards Response Time, biased towards Performance, biased towards Response Time and Performance, and so on. For this purpose, we propose to define different sets of inference rules. The first set consists of unbiased rules, i.e. rules which treat all parameters as equal. The other sets of rules consider one parameter each as more important than other parameters. Using different sets of rules the estimated rating will be different, for e.g. the estimated rating obtained by using unbiased set of rules will give an unbiased rating which will be different from that obtained by using a set of rules biased towards a particular attribute (e.g. response time). The rating given by the user is compared against all
of the estimated ratings obtained through the fuzzy inference and the estimated rating which is the closest match is considered equivalent to the user's rating. Then the bias used for getting that estimation will be established as the bias the user held when rating the service. If the user's rating does not match any of the estimated ratings, then the users rating is termed invalid because the estimated ratings are calculated for each possible case.

Following is the mapping of the problem into the fuzzy inference process. We have used Mamdani type fuzzy inference because our output Rating is non linear.

3.2.1 Input Membership Functions

The inputs to the fuzzy system are the compliance values in each attribute. For each input (attribute), 3 fuzzy sets (correspondingly, 3 input membership functions) have been defined, so 3 levels of compliance are identified, namely: low, compliant and high. Membership functions “compliance is low” and “compliance is high” are represented as trapezoidal functions. The “service is compliant” membership function is represented as a triangular function. The number of fuzzy sets (and hence membership functions) used depends on the level of differentiation desired and may be different for different attributes. The type of function used depends on the input entity (the attribute). We have chosen the input membership functions as trapezoids and triangular because these are the simplest functions that capture the properties of our input, i.e. compliance values. The shape of the function depends on the parameter (attribute) for which the compliance level is calculated. Different attributes may have different shapes of the functions. The shapes we have used are exemplars/illustrations and are only meant as guides.

Compliance calculation is based on the following formula (refer to section 3.1)

\[ C_j = (a_{dj} - a_{pj}) / a_{pj} \]

where \( a_{dj} \) = delivered value of attribute \( a \) in \( j^{th} \) invocation
\( a_{pj} \) = projected value of attribute \( a \) in \( j^{th} \) invocation

Since the projected and delivered values of an attribute \( (a_d \text{ and } a_p) \) are unbounded and depend on the attribute itself, it becomes difficult to determine the range for the compliance values. Hence in our model, we have assumed bounds such that the compliance values are in the range of \([-1 \text{ to } 1]\). Compliance values of greater than 1 are considered to be 1, i.e. they are all high compliance irrespective of the value itself. On the lower bound compliance calculation formula itself limits the bound at -1. Figure 1 shows the general shapes of the different input membership functions.

3.2.2 Output Membership Functions

The output of the fuzzy inference process is the estimated rating value. We have identified 4 output fuzzy sets (correspondingly, 4 output membership functions) for “rating”, namely, poor, moderate, good and excellent. These are also of trapezoidal and triangular types. Figure 2 shows the general shapes of the output membership functions for “rating” ranging from 0 to 10. The shapes we have chosen are again exemplars/illustrations and are only meant as guides.

3.2.3 Fuzzy Inference Rules

Fuzzy inference rules are defined for relating inputs to the outputs. In this case, fuzzy inference rules relate the compliance values in different attributes to an estimated rating value. Rules are defined on the notion that if a service/provider performs well in all or most of the attributes, it should be given a high rating and vice-versa. Figure 3 presents unbiased fuzzy inference rules for 3 level “compliance” inputs and 4 level “rating” outputs.

For estimation of biased ratings, sets of biased inference rules are used. Each of these sets of rules is used to calculate a rating for different weights of the parameters. Here weights represent the logical weights, or the importance a
In the next section, we present some illustrative scenarios to demonstrate how our approach (fuzzy estimation of rating) can be used to identify user bias and validate ratings.

![Figure 4: Biased fuzzy inference rules](image)

4. ILLUSTRATIVE SCENARIOS DEMONSTRATING HOW THE FUZZY INFERENCE ENGINE INFERS RATIONALE FOR USER RATINGS

In this section, we provide illustrative scenarios validating our hypothesis that analyzing user ratings along with performance of services provides an improved insight into user’s rating behavior and facilitates in identifying user preferences and detecting deceitful users. To support our hypothesis we formulate and analyse several scenarios which establish that rationale for user ratings can be inferred through comparison of ratings provided by the user with the performance of the services. In order to estimate ratings based on services’ performance, we implemented a fuzzy inference engine using MATLAB 7.0 as per the mapping discussed in section 3.2. The estimated ratings are then compared with the rating provided by the user to determine the rationale for the (user) rating. Thus inferred rationale can then be used for various purposes as discussed in section 2.3.

Next we describe our experimental setup for the fuzzy inference engine and subsequently present scenarios simulated with test data to practically demonstrate our arguments.

Our fuzzy model allows for any number of quality attributes to be used in the estimation of ratings, however, for the purpose of our demonstration, we have used only three attributes, namely, Response Time, Availability, and Performance.

Similarly, for simplicity, we have used a 3-level categorization in input values (compliance) - low, compliant, high and 4-level categorization in output values (rating) poor, moderate, good, excellent. Categorization levels can be increased as per the granularity desired.

### 4.1 Identifying Invalid User Ratings

In each of the following cases, the table shows agreed and delivered values of the three SLA parameters: Response Time, Availability and Performance and corresponding compliances. Taking these compliances as the input our fuzzy inference engine calculates the estimated ratings for four cases: unbiased, Response Time-biased, Availability biased and Performance biased. Assuming the ratings (both user ratings and estimated ratings) to be in the range of 1 to 5 (where 1 is lowest and 5 is highest), we show how our inference engine works in each case.

In the case shown in Table 1, the user rating 4.0 (excellent) matches the estimated ratings (excellent). Hence the inference engine will identify this rating as valid. When we consider the performance details, it is seen that this inference is correct because the service performed well in all the parameters and the user gave a high rating 4.0 (excellent), which is reasonable/valid.

In the next case, shown in Table 2, for the same performance statistics, the user gave a rating of 2.0 (moderate). In this case, the user rating will be identified as invalid because the user’s rating 2.0 (moderate) does not match any of the estimated ratings (excellent). This inference that the rating is invalid is correct because the user has given a low rating even though the service performed well in all parameters. Hence in this case, rationale could be deception.

In the case shown in Table 3, the user’s rating is identified as invalid because it, 3.0 (good), does not match any of the estimated ratings (poor). As can be seen from the performance details, the service performed poorly in all the

### Table 1: Validating user ratings

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Agreed value</th>
<th>Delivered value</th>
<th>Compliance</th>
<th>User Rating</th>
<th>Estimated Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>10</td>
<td>8</td>
<td>0.20</td>
<td>4.0</td>
<td>4.61, 4.62, 4.06, 4.65</td>
</tr>
<tr>
<td>Av</td>
<td>90</td>
<td>100</td>
<td>0.11</td>
<td>2.0</td>
<td>4.61, 4.62, 4.06, 4.65</td>
</tr>
<tr>
<td>P</td>
<td>75</td>
<td>95</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Detecting deception

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Agreed value</th>
<th>Delivered value</th>
<th>Compliance</th>
<th>User Rating</th>
<th>Estimated Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>10</td>
<td>8</td>
<td>0.20</td>
<td>2.0</td>
<td>4.61, 4.62, 4.06, 4.65</td>
</tr>
<tr>
<td>Av</td>
<td>90</td>
<td>100</td>
<td>0.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>75</td>
<td>95</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RT: Response Time, Av: Availability, P: Performance

### Table 3: Unbiased ratings

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Agreed value</th>
<th>Delivered value</th>
<th>Compliance</th>
<th>User Rating</th>
<th>Estimated Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>10</td>
<td>8</td>
<td>0.20</td>
<td>4.0</td>
<td>4.61, 4.62, 4.06, 4.65</td>
</tr>
<tr>
<td>Av</td>
<td>90</td>
<td>100</td>
<td>0.11</td>
<td>2.0</td>
<td>4.61, 4.62, 4.06, 4.65</td>
</tr>
<tr>
<td>P</td>
<td>75</td>
<td>95</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RT: Response Time, Av: Availability, P: Performance
parameters, still the user gave a high rating, which means this rating is unreliable. Hence our inference that the rating is invalid is correct. In this case, the rationale could be collusion.

### 4.2 Establishing User Bias

In the following examples also, in each case the table shows the agreed and delivered values of the three SLA parameters: Response Time, Availability, Performance; corresponding compliances and estimated ratings for the 4 cases.

In the case shown in Table 4, the user’s rating 3.5 (good) matches the estimated Response Time-biased rating 4.0 (good); the system will infer that this user is biased towards response time. When we consider the performance statistics, it is seen that the service performed well in only response time, still the user has given a high rating. Hence the inference that the user is biased towards response time is correct. Hence, in this case, the rationale is bias towards a certain attribute (response time). However, biasedness cannot be confirmed by examining a single invocation. If biasedness is detected in one invocation, then the user’s other invocations are also examined to establish the bias.

In the next case, (Table 5) assuming that the system has previously established from the user’s rating behavior that the user is biased towards response time; the user’s rating 3.5 (good) does not match estimated Response Time-biased rating 0.633 (poor), hence the inference engine will identify this rating as invalid. This inference is reasonable because the systems knows that the user is biased towards response time, but the user gave a high rating when the service performed badly in response time, hence this rating is definitely invalid. Hence, in this case, the rationale could be collusion.

### 5. INCORPORATING RATIONALE INFERENCE IN EXISTING WEB SERVICES ARCHITECTURE

In this section, we explain how the inference engine can be incorporated in existing web services operational framework.

The architecture for SLA enabled web services typically consists of three components: the service provider, the service broker and the end user [8]. The service provider publishes SLA enabled web service(s) and sends it to the service broker for storage in a repository. An end user registers with the service broker, searches the repository and selects a suitable service provider. The provider and the end user negotiate the relevant parameters and finalize the service level agreement (SLA). This point onwards, the transaction is monitored by a third party [5], (Performance Monitor), which verifies the values of SLA parameters in the agreement (SLA) against the values obtained by probing or intercepting the client invocation. Delivered values thus obtained through interception are then compared with the agreed values in the SLA to calculate the compliance of the service [4] (Compliance Calculator).

Having access to both the compliance of the provider and the rating given by the user makes it possible to have a comparison of the user rating with the performance of the service/provider to find how consistent the rating is with respect to the performance. This gives an insight into the user’s rating behavior, which enables the system to infer the rationale for the ratings (Fuzzy Reasoning Engine), which in turn facilitates in detecting deception and collusion (and subsequently identifying invalid ratings) and inferring user preferences. Thus elucidated rationale can then be used to make informed evaluation of the services/providers (R eputation Evaluator). Figure 5 shows the proposed architecture incorporating our fuzzy reasoning engine.

Through filtering of invalid ratings, this model enhances
the functionality of the Reputation System, making it robust from attacks by malicious entities. Also inclusion of rationale along with ratings makes it possible to personalize and tailor reputation evaluations to particular user requests, subsequently enhancing the functionality of the service broker.

Existing reputation systems which employ aggregation of user ratings as the basis for reputation assessment can make use of our fuzzy inference engine to filter out invalid ratings, and hence can be incorporated as the Reputation Evaluator in our model. Also, the reputation models that take user preferences into account while assessing reputation can make use of our fuzzy reasoning engine to infer user preferences instead of asking users explicitly and hence can also be incorporated as the Reputation Evaluator in our model. However, to fully exploit the benefits of our reasoning engine, the reputation evaluator needs to have mechanisms to incorporate rationale into its reputation calculations, which is currently unavailable. Hence our next step is to design and develop a reputation evaluator which incorporates rationale inferred by the fuzzy reasoning engine into its reputation calculations, so that the system can deliver personalized and customized reputation assessment as per the user request.

6. CONCLUSIONS AND FUTURE WORK

We have established that analyzing user rating behavior against service performance gives insight into rationale for user ratings. We have also established that explicating rationale can be very useful in detecting and eliminating invalid ratings and identifying user preferences, which facilitate in preventing collusion and deception and hence make the reputation system more robust to the attacks of malicious entities. Also inferring user preferences facilitates in making personalized reputation evaluations. We have proposed a fuzzy logic based approach for inferring rationale for user ratings. We have shown through our prototype implementation that this model does detect invalid (deceptive and collusive) ratings and infer user preferences.

At this stage the model can only detect user preference for a single attribute and can identify ratings as valid or invalid. It would be desirable to enhance this model to make it more accurate as to be able to identify whether the user has been too harsh or too lenient in rating the service. We are working towards improving our existing model to be able to detect subtle variances (leniency and harshness) in user rating behavior and adjust ratings accordingly. Also, we plan to explicate the inferred rationale so as to make it possible to transfer rationale along with rating/reputation in a distributed reputation management environment. Furthermore, currently this model can only infer user preferences and identify invalid ratings; in future we intend to develop a system that utilizes this inferred rationale for purposes such as detecting collusion and giving personalized recommendations. We are also working towards developing a reputation evaluator (refer to the discussion at the end of section 5), which incorporates rationale inferred by the fuzzy reasoning engine into reputation calculations.

7. REFERENCES