Decomposing Ratings in Service Compositions

Icamaan da Silva[1] and Andrea Zisman[2]

[1] Department of Computer Science, City University London, United Kingdom
icamaan.silva.1@city.ac.uk

[2] Computing Department, The Open University, United Kingdom
Andrea.Zisman@open.ac.uk

Abstract. An important challenge for service-based systems is to be able to select services based on feedback from service consumers and, therefore, to be able to distinguish between good and bad services. However, ratings are normally provided to a service as a whole, without taking into consideration that services are normally formed by a composition of other services. In this paper we propose an approach to support the decomposition of ratings provided to a service composition into ratings to the participating services in a composition. The approach takes into consideration the rating provided for a service composition as a whole, past trust values of the services participating in the composition, and expected and observed QoS aspects of the services. A prototype tool has been implemented to illustrate and evaluate the work. Results of some experimental evaluation of the approach are also reported in the paper.

Keywords: Rating decomposition, rating propagation, trust values, feedback.

1 Introduction

In a highly competitive environment where anyone can become a service provider and the number of similar services available increases quickly, it is crucial for a system to be capable of choosing the most suitable service for a particular user. Trust and reputation have been the focus of research in several open systems such as e-commerce, peer-to-peer, and multi-agent systems [5][10][11]. Some trust and reputation approaches have also been suggested for web-service systems [7][13][14], and have been used in several e-marketplaces applications such as eBay [2], GooglePlay [4], and Amazon [1]. In general, trust and reputation web-services based approaches are limited and immature [14]. For example, these approaches (i) assume that information given by service providers can be trusted; (ii) assume that feedbacks provided can always be trusted; (iii) demand a large number of interactions or non-intuitive information from users; and (iv) do not properly handle the existence of malicious users when considering their feedback.

An important feature of service-based systems is the fact that services are formed by the composition of other services and in many situations the existence of several services is transparent for service consumers; i.e., service consumers do not know if they are using a single service or a composition of services. In this context, service consumers normally provide feedback to the composition as a whole without considering that the service is composed of several resources. The participating services in a composition, and the way they interact with each other, may influence the feedback associated with the composition. When creating new service compositions it is necessary to distinguish between “good” and “bad” services and to consider the reputation of the individual services. In a competitive market, service providers should also know about the reputation of their services to improve them.

It is essential to have ways to decompose provided ratings and trust values of a composition to the individual services in the composition. However, an approach in which a rating given for a composition is replicated to, or averaged with, the services participating in the composition is not appropriate since it will not provide fair ratings to the participating services. For example, some participating services that performed well may be penalised by other services in the composition that performed badly.

In this paper we present a framework to support the decomposition of service ratings to individual services participating in a service composition. The framework uses a rating decomposition approach that considers (i) rating provided by a user to a service composition as a whole, (ii) previous trust values associated with the individual services in the composition, (iii) the values of QoS aspects that the individual services took to perform their tasks (observed QoS values), and (iv) the QoS values specified for the services in the composition by their respective service providers (expected QoS values). The previous trust values associated with individual services are calculated based on a trust model that we have previously proposed [12].

Motivating Example. As an experiment to illustrate how the decomposition process impacts on the trust values of the services in a composition, we present in Table 1 an example in which a user provides ratings (*R*) in different intervals to a service composition with two services *s1* and *s2*. Assume the initial trust values associated with *s1* and *s2* as 0.7 and 0.3 respectively. We run the experiment for 25 interactions. Table 1 shows the final trust values associated with *s1* and *s2* after the 25 interactions using an approach as the one we are suggesting and an approach in which the ratings are replicated. The results presented in the table show that replicating the ratings provided to the service composition to the participating services tend to penalize the services with higher trust values and favour the services with lower trust values. After the 25th interaction the trust values associated with *s1* and *s2* are nearly the same when we replicate the ratings provided to the service composition. This is not the case when using the approach described in this paper, since the different rating values calculated to the individual services provide distinct new trust values to the services.

Table 1. Trust values of services with s1 and s2 for different decomposition approaches

	Our approach		**Replication of *R***	
Ratings provided	Trust Values – ***s1***	Trust Values – ***s2***	Trust Values – ***s1***	Trust Values – ***s2***
[0.0, 2.5[	0.20	0.09	0.14	0.12
[2.5, 5.0[	0.53	0.23	0.39	0.36
[5.0, 7.5[	0.86	0.37	0.59	0.57
[7.5, 10.0]	0.98	0.61	0.85	0.82

The rest of this paper is structured as follows. In Section 2 we present our rating decomposition process. In Section 3 we discuss implementation and evaluation aspects. In Section 4 we give an account of related work. In Section 5 we present final remarks.

2 Rating Decomposition

In this section, we describe the mechanisms used to decompose a rating *R*, provided by a user, to a service composition, into ratings associated with individual services participating in the composition. We also present a trust model to calculate trust values of individual services participating in a composition.

Our framework deals with service compositions that are transparent to the users. This means that users of a particular service composition do not distinguish whether they are accessing a composition of services or only a single service component. The users are not able to provide ratings to the individual services in a composition or to specify different levels of importance to the individual services in a composition.

The rating decomposition approach used in our framework considers (i) the rating provided by a user U to a service composition, (ii) the previous trust values associated with the participating services, (iii) the observed QoS values of the participating services, and (iv) the expected QoS values specified for the participating services by their respective service providers. For illustrative purpose, in this paper we concentrate on response times QoS values. In the approach, the values for ratings associated with services in a composition are within the interval [0.0, 10.0], as are the ratings provided for a service composition by the users. More specifically, the decomposed rating for a service s_i in a composition is given by the equation below:

$$r(s_i) = R \times \frac{T(s_i)}{\frac{\sum_{l=1}^{n} T(s_l)}{n}} + \frac{\sum_{j=1}^{m} p(s_i, t_j, t'_j)}{m} \tag{1}$$

with

$$p(s,t,t') = \begin{cases} 0.0 \quad if \quad -0.1 \leq \frac{t'-t}{t'} \leq 0.1 \\ -1.0 \quad else\ if \quad \frac{t'-t}{t'} \leq -0.5 \\ 1.0 \quad else\ if \quad \frac{t'-t}{t'} \geq 0.5 \\ \frac{t'-t}{t'} \times 2 \quad otherwise \end{cases} \tag{2}$$

where:

- s_i: is a service participating in a service composition;
- $r(s_i)$: is the final rating calculated for service s_i;
- *R*: is the rating provided by a user for the service composition;
- $T(s_i)$: is the trust value calculated for service s_i;
- *n*: is the number of component services in the service composition;
- *l*: is an index representing all the services in the composition ($1 <= l <= n$);
- *p(s, t, t')*: is a recompense function (penalty score) calculated for a service *s* based on the QoS value of *s* to perform its task and the QoS value specified for *s*;
- *t'*: is the QoS specified for s_i by the service provider (expected);
- *t*: is the actual QoS value that s_i took to perform its task (observed);

- m: is the number of considered QoS values;
- j: is an index representing all the considered QoS values ($1 <= j <= m$).

We use trust values of the participating services in a composition to calculate the individual ratings of the services in order to analyse how well a service has performed when compared to the other services participating in the composition. We assume that a service with a high trust value has performed well in the past, while a service with a low trust value has performed poorly. Nepal, *et al*. [9] believes that taking into consideration past trust values of participating services when decomposing a rating offers a certain level of consistency. In his view, when a service has performed better than other services in the past, then this service tends to continue to perform better.

As shown in equation (1), the approach considers a *recompense function* ($p(s,t,t')$) during the rating decomposition process. As the name suggests, the *recompense function* is intended to reward a service in case its performance is better than what it was stated by its service provider (in terms of QoS values), or penalize the service otherwise. For a service s_i, the function has as input parameters the QoS values stated by the service provider, and the actual QoS value that the service took to perform a task when it was invoked. Positive values for the recompense function signify a reward to the service, while negative values signify a penalty to the service.

Given that several aspects may cause variations on the QoS values of a service (e.g., number of requests at a time, quality of the network connection), we have limited the possible result values for this function. This is to prevent the recompense function to cause a high influence in the rating decomposition process, since the function is only intended to reward or penalize a participating service. For example, the highest possible value as output is 1.0 (similarly a penalty of -1.0) when the difference between the actual QoS value is at least 50% lower than the QoS value stated by the service provider (similarly when the actual QoS value is at least 50% higher than the one stated by the provider). We also consider that small variations between the actual QoS value of a service and the QoS value stated by the service provider should not be rewarded or penalized. In this case, we consider a difference of 10% between the actual and given QoS values as being a small variation. The values above were identified after running some experiments with different variations.

Trust Value Calculation. The decomposition of ratings relies on trust values associated with the services in a composition. In the approach, trust values are calculated based on past ratings identified for the participating services. The trust values associated with a service are values in the interval [0.0, 1.0]. In the case in which a service s does not have associated past ratings to calculate the trust value of s, the approach assumes a trust value of 0.5 for s. This value represents the average of possible rating values. The trust model used to calculate the trust values associated with the participating services is based on the trust model we have described in [12].

The calculation of trust values is based on the Dirichlet probability distribution expected value [8]. The ratings given to the composition and decomposed into the participating services are continuous values between 0.0 and 10.0. Each rating calculated for a service is mapped into a 5-component variable ($v_1, \dots, v_5$) based on the calculation of the level of membership ($m(c, v_i)$) of a continuous rating, according to the equation described by Josang *et al*. [6]. The levels of memberships of the 5-component variable are represented as a vector of size five ($\vec{V}$). For example, consider the situation in which the decomposed rating of a service is 7.0. In this case, the membership vector ($\vec{V}$) would be calculated as [0, 0, 0.2, 0.8, 0].

To calculate the trust value associated with a particular service the membership vectors are aggregated through a weighted sum. In order to weight each rating (membership vector) an aging factor component is used. The aging factor is intended to give more importance to recent ratings than old ones. As defined in [5], the trust value of a service based on ratings is calculated by the function below:

$$T(s) = \sum_{j=1}^{k} \rho_j \delta_j \qquad \text{with} \qquad (3)$$

$$\rho_j = \frac{(j-1)}{(k-1)} \qquad \delta_j = \frac{\vec{R}[j]+C}{\sum_{m=1}^{k}(\vec{R}[m]+C)} \qquad \vec{R} = \sum_{l=1}^{n} \vec{V}_l \alpha^{\Delta t} \qquad (4)$$

where:

- $\vec{R}$: is the aggregated vector calculated by the weighted sum of all the vectors $\vec{V_l}$;
- $\vec{V_l}$: is the membership vector mapping a decomposed rating to a service *s*;
- *n:* is the total number of ratings decomposed for the participating service *s*;
- *k*: is the size of $\vec{V}$ (k=5);
- ρ_j: is a value assigned to each component *v1,…,vk* to give a value in an interval;
- *C*: is a constant used to ensure that all values in the elements of the vector are greater than 0, to allow a posterior analysis of the Dirichlet distribution;
- $\alpha^{\Delta t}$: is the aging factor, where α is a constant and Δt is the difference in terms of time between the available ratings for *s* ($\Delta t \in \mathbb{N}$, $\mathbb{N}$ is the set of natural numbers).

In order to illustrate, consider the scenario in which a participating service has three available ratings to calculate its level of trust: 7.0, 8.0, and 6.0. The membership vectors for the three ratings are $\vec{V}_1$= [0, 0, 0.2, 0.8, 0], $\vec{V_2}$= [0, 0, 0, 0.8, 0.2], and $\vec{V}_3$= [0, 0, 0.6, 0.4, 0]. Then, applying equations (4) the aggregated vector is $\vec{R}$ = [0, 0, 0.8, 2.0, 0.2] (considering Δt = 0, which means all received ratings were received in the same period of time). Finally, the trust value associated with the participating services is *T(s) =0.67* (considering *C* = 0.1).

3 Implementation Aspects and Evaluation

A prototype tool has been implemented in order to evaluate the main aspects of the approach. The tool has two main modules, namely (i) rating decomposition module and (ii) trust calculation module. To evaluate the approach, we also implemented a simulator to generate ratings for the evaluation. The prototype and simulator were implemented using Java. The proposed approach has been evaluated in terms of two different cases: *case (1) - the impact that the ratings provided for a composition have in the rating decomposition process; case (2) - the impact that the observed QoS values of the services have in the rating decomposition process.*

Case (1): In this case we want to evaluate the effects that the trust values associated with the participating services in a composition have in the rating decomposition process, as well as the impact that the ratings provided for the composition have in the rating decomposition process. More specifically, we want to analyse how the rating decomposition process considers services with different trust values and different given service composition ratings. This experiment is important because service compositions may be formed by services with different trust values, which need to be considered when decomposing a rating provided for a whole composition.

In the experiments we considered a service composition with four services (*s1, s2, s3,* and *s4*) with trust values of 0.26, 0.50, 0.74, and 0.98, respectively. These values provide an average trust value of 0.62. We assumed that the trust values were calculated based on a history of ten ratings previously decomposed for each of the participating services. In the experiments we also considered 25 units of times (time-steps), and that the service composition received ratings are based on a uniform distribution in every time-step. We also considered response times as the QoS aspects and that the difference between the expected and observed response times of a service is less than or equal to 10% (i.e., recompense function $p = 0.0$, as per equation (2)).

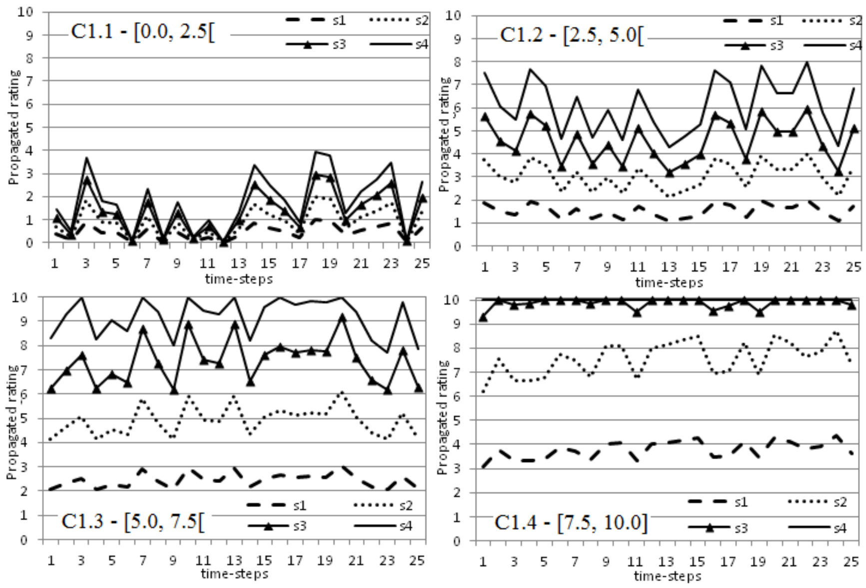

Fig. 1. Propagated ratings to participating services (Case (1))

We executed the experiments for four different cases (C1.1, C1.2, C1.3, and C1.4), differing on the interval of ratings provided to the service composition in each time-step. Case C1.1 considers ratings provided to the service composition in the interval [0.0, 2.5[; while cases C1.2, C1.3, and C1.4 consider ratings in the intervals [2.5, 5.0[, [5.0, 7.5[, and [7.5, 10.0], respectively. We measure the rating decomposed (propagated) for each service in each time-step using our decomposition process. Fig. 1 shows the results of the experiments for the four cases C1.1 to C1.4 above.

As shown in Fig. 1, in all cases, there is an oscillation in the ratings decomposed for the services. The results also show that in cases C1.2 and C1.3 the curves for the decomposed ratings have a similar behaviour for all the four services. In case C1.4, the values of the decomposed ratings for services *s3* and *s4* are similar since the services cannot have ratings higher than 10.0. In case C1.1, when the rating provided for the composition is close to 0.0, the decomposed ratings are also quite similar.

Case (2): In this case we want to analyze the effect of the recompense function in the rating decomposition process. More specifically, we are interested in the comparison on how the expected and observed QoS values for services participating in a composition can influence the ratings decomposed to these services. This experiment is important because service compositions may be formed by services that performed in different ways and, therefore, these services need to be penalised or rewarded.

In the experiments we considered a service composition with five services (*s1, s2, s3, s4,* and *s5*) and the response time as the QoS aspect. We assume that all the participating services have not received previous ratings and, therefore, they have the same trust values (0.5) in the first time-step. We consider the five services with different probabilities of exceeding the expected response times. These probabilities are 0%, 25%, 50%, 75%, and 100% for services *s1, s2, s3, s4,* and *s5* respectively. For example, while service *s5* will always exceed its expected response time, services *s2* and *s3* will exceed their expected response times in 25% and 50% of the cases respectively, and service *s1* will never exceed its expected response time. Given that we are considering observed response times that exceed the expected response times, the values for p will be between [-1.0, 00] (see equation (2)). In the experiments, the values of the exceeded expected response times for the participating services are based on a uniform distribution in the interval [0.1, 0.5] of the exceeded percentage value (e.g., 0.1 means that the component service exceeded its expected time in 10%).

Similarly to Case 1, we considered 25 units of times (time-steps), and that the service composition receives ratings based on a uniform distribution in every time-step. We executed the experiments for four different cases (C2.1, C2.2, C2.3, and C2.4), differing on the interval of ratings provided to the service composition in each time-step. Case C2.1 considers ratings provided to the service composition in the interval [0.0, 2.5[; while cases C2.2, C2.3, and C2.4 consider ratings in the intervals [2.5, 5.0[, [5.0, 7.5[, and [7.5, 10.0], respectively. For each case, we measure the rating decomposed (propagated) for each service in each time-step using our decomposition process. Fig. 2 shows the results of the experiments for the four cases.

As shown in the Fig. 2, after a few numbers of time-steps, except for service *s1*, the curves of the decomposed (propagated) ratings are very close to each other, making it hard to differentiate the one with higher rating. This is due to the fact that any service can get a higher decomposed rating since, in this experiments, (a) we do not differentiate the past trust values of the services and (b) the probability of a service exceeding its response time does not interfere with the differences between the observed and expected response times. For example, although *s5* is always exceeding its expected response time, the difference between the expected response time and the observed response time can be small; while service *s2* that exceeds its expected response time in only 25% of the cases could have an observed response time much higher than its expected one, which will cause *s5* to have a higher decomposed rating than *s2*. Even service *s1* could receive decomposed ratings similar to the other services, as it is in case C2.1 for the first and second time-steps, in which services *s1, s2,* and *s3* have the same decomposed ratings.

Another point to be highlighted is the fact that the results in the experiments in Case 2 show that the differences between the decomposed ratings of the services are smaller than the differences of the decomposed ratings of the services in Case 1. This is due to the fact that in Case 2 the trust values of the services are not being considered - all the services have the same past values of 0.5. Moreover, the trust values of the services have a bigger impact in the decomposed ratings than the penalty (or reward) given for the expected and observed response times.

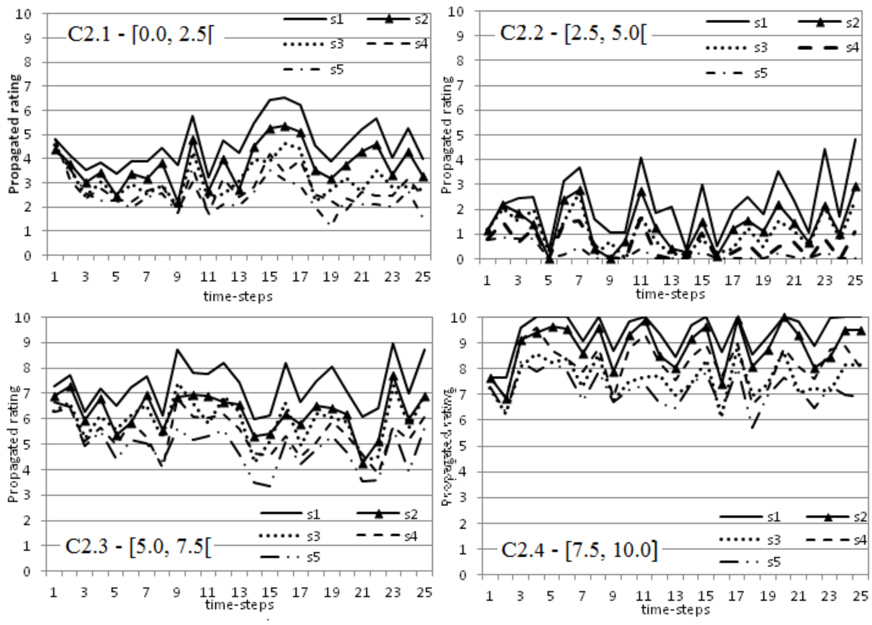

Fig. 2. Propagated ratings to participating services (Case (2))

4 Related Work

Several approaches have been proposed to support service selection, trust, and reputation management over the last years [5][9][10][11]. Most of these approaches focus on reputation management aspects [5][10][11]. Very few approaches consider the fact that services are composed by other individual resources (services) and reputation scores need to be reflected in the individual services [9].

Although rating decomposition is an area that has been investigated in different disciplines such as Business, Cognitive Psychology, Mathematics, and Computer Science, there are few approached that supports rating decomposition in service-based systems [3][9]. Nepal et. al. [9] propose a methodology to propagate ratings provided to a service composition into participating services. Similarly, the approach takes into account the trust values of the participating services. However, it assumes that service consumers are aware of the participating services in a composition. Goldberg et. al. [3] propose an approach based on Singular Value Decomposition (SVD) technique in which rated objects are represented as latent variables that allow discriminating between positive and negative ratings. This technique is effective to predict the user appreciation of an object, but it does not provide ways of discriminating the resource that affects the rating of the whole object. Srivastava and Sorenson [13] describe an approach to service selection based on user's perception of the QoS attributes, rather than the actual attribute values. They propose an interactive approach to find out the most appropriate values for each QoS attribute. The framework and process described in this paper complement existing trust-based approaches by providing a way of decomposing ratings given to a whole composition into ratings for individual services, and considering past trust values, and observed and expected QoS values of the individual services in a composition.

5 Final Remarks

We describe a framework that considers ratings provided for a composition as a whole and decomposes this rating based on past trust values of the services in the composition, as well as expected and observed QoS aspects of the services. The decomposed ratings of the participating services are also used to calculate new trust values for the services based on a trust model approach that we have developed. We are currently extending the framework to consider ratings received by a service when this service is invoked in isolation, together with the rating received for a composition in which the service is participating. We are expanding the framework to consider dependencies between services in a composition, since a service can have different ratings depending on how well it interacts with other services in a composition.

6 References

[1] Amazon. http://amazon.com.

[2] eBay. http://ebay.com.

[3] K. Goldberg et al. Eigentaste: A Constant Time Collaborative Filering Algorithm Information retrieval, Volume 4, Number 2, 2001.

[4] GooglePlay. http://googleplay.com.

[5] A. Josang and J. Haller. Dirichlet Reputation Systems. 2nd International Conference on Availability, Reliability and Security (ARES 2007), Vienna, April 2007.

[6] A. Josang, X. Luo, and X. Chen. Continuous Ratings in Discrete Bayesian Reputation Systems. Proceedings of the IFPIM 2008.

[7] Y. Liu, A. Ngu, and L. Zheng. QoS computation and policing in dynamic web service selection. Proc. of World Wide Web Conference, 2004.

[8] J. I. Marden. Mathematical Statistics: Old School. University of Illinois, 2012.

[9] S. Nepal, Z. Malik, and A. Bouguettaya. IEEE Int. Conf. on Web Services ICWS, 2009.

[10] S. Ruohomaa and L. Kutvonen. Trust Management Survey. Proc. of iTrust, 2005.

[11] J. Ben Scharfer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative Filtering and Recommender Systems. Lecture Notes in Computer Science, V. 4321, 2007.

[12] I. B. V. Silva and A. Zisman. A Framework for Trusted Services. International Conference on Service Oriented Computing ICSOC, 2012

[13] A. Srivastava and P. G. Sorenson. Service Selection based on customer Rating of Quality of Service Attributes. IEEE International Conference on Web Services, 2010.

[14] Y. Wang and J. Vassileva. Towards Trust and Reputation Based Web Service Selection: A Survey. International Transaction Systems Science and Applications, v. 3, n. 2, 2007.