A trust and reputation model based on bayesian network for web services

Hien Trang Nguyen
Macquarie University

Weiliang Zhao
University of Wollongong, wzhao@uow.edu.au

Jian Yang
General Research Institute for Non Ferrous Metals,Ministry of Science & Technology, China,Macquarie University

2010 IEEE International Conference on Web Services

A Trust and Reputation Model Based on Bayesian Network for Web Services

Hien Trang Nguyen, Weiliang Zhao, Jian Yang
Department of Computing
Macquarie University
Sydney, Australia
{*hinguyen, wzhao, jian*}*@science.mq.edu.au*

***Abstract*—Trust and reputation for web services emerges as an important research issue in web service selection. Current web service trust models either do not integrate different important sources of trust (subjective and objective for example), or do not focus on satisfying different user's requirements about different quality of service (QoS) attributes such as performance, availability etc. In this paper, we propose a Bayesian network trust and reputation model for web services that can overcome such limitations by considering several factors when assessing web services' trust: direct opinion from the truster, user rating (subjective view) and QoS monitoring information (objective view). Our comprehensive approach also addresses the problems of users' preferences and multiple QoS-based trust by specifying different conditions for the Bayesian network and targets at building a reasonable credibility model for the raters of web services.**

***Keywords*-Web service; trust; reputation; service selection; Bayesian network;**

I. Introduction

Web service selection is one of the challenges in Service Oriented Computing (SOC). Actually, selecting the best web service is equivalent to selecting the most trusted web service with the highest ranking level. Hence recently many studies focus on adopting trust and reputation technology as a solution for web service selection problem. A good trust and reputation model for web services should consider the following requirements:

1) The trust should cover multiple aspects of QoS of web service as among a bunch of web services offering similar functionality, consumers need to know not only the function of the service but also the qualities of the service.
2) The trust should be determined based on user preferences because different consumer maybe interested in different quality attributes and a service suitable for one user may not be suitable for another [11].
3) It is necessary to combine both subjective dimension (users' feedback) and objective dimension (QoS performance monitoring) to assess the trust and reputation of web services. Reputation is originally a subjective conception, which predicts future behavior of an entity based on past behaviors; hence users' rating is a very important factor for consumers to share their knowledge of direct experiences in interacting with the web services. However, beside subjective users' feedback, objective performance measure is also a significant dimension because it reflects web services' ability to conform to the Service Level Agreement (SLA) and to deliver the promised QoS parameters, such as availability, performance. Moreover, trust based on QoS monitoring is especially useful when users' ratings suffer from common problems, for example dishonest rating, spamming, etc.
4) Rater credibility should be considered seriously in the trust assessment process so as to indentify how much a requester can rely on the other users' ratings. Without this, the trust and reputation system easily fails because of false ratings, collusion and deception problems.

Recently, many trust and reputation models for web services have been proposed such as [1], [2], [5], [8], [9], [6], [11], [7], [10]. Each approach takes different sources of trust and different computation methods to derive the trust values, however, rarely a solution can satisfy all of the four requirements above. In this paper, we propose a trust and reputation model based on Bayesian Network for web services that can not only satisfy users' preferences on multiple QoS attributes but also integrate different sources, i.e. users' ratings, QoS monitoring measure and direct experience opinion of the requester. Besides, we present a mechanism that specifies how consumers should rate web services and a model for computing rater credibility.

The rest of the paper is organized as follows: Section II discusses some related work and their limitations. Section III explains our trust and reputation model and metrics. Next, the calculation algorithms are described in section IV. Section V presents a running example and finally the paper ends with concluding remarks and future work.

II. Related Work

In this section we review some trust and reputation models for web services in terms of their sources of trust and their computation methods.

Recommendation is the most popular source of trust in the current models such as in [1], [2], [5], [8]. In [1], the authors propose a personalized model targeting multiple QoS where the trust is derived by calculating the moment difference

978-0-7695-4128-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ICWS.2010.36

between the consumers' preferred quality attribute value and the vector (minimum, maximum, typical, aggregated reputation values obtained from ratings) of the quality attribute. However this model does not consider rater credibility. In [2], the reputation is the sum of several factors: the rating value, the transaction price and the rater credibility; yet it is an impersonal model where service reputation is unique in the view of all consumers. The personalized models are more desirable to satisfy different user's preferences.

A few models consider direct experience opinion of the requesting consumer itself, gained from past interactions. The requester itself gives the service ratings. Although this is similar to the feedbacks provided by third parties, it is a more important source and should be weighted more. An example of this type is in [3]. It uses collaborative filtering technique to predict the rating from user u to service i, by firstly identifying the set of all web services that user u has rated, then comparing each service j in the set with the target service i. This kind of *local trust* model (the requesting consumer only considers the opinion of itself) is insufficient to assess web services.

QoS monitoring information is another type of trust source. Some QoS attributes can actually be measured by some engines during transactions, for example the response time, the availability, etc. [9] proposes to assess the reputation of web services based on the attribute compliance - the difference between the projected and the delivered values of quality. Yet, this is the only source of trust in this model and it applies a simple classical statistics method for computation (average and variance).

Some models such as [6], [11], [7], [10] combine more than one source of trust. [6] presents a complicated method to rank web services based on different quality attributes, through many steps: (i) it clusters raters into groups then weigh groups based on group size and stability, (ii) it combines the groups' reports about conformance level of a quality attribute (iii) it uses users' preferences on QoS to integrate the rankings of all quality attributes to derive a final score for web services. Basically this model satisfies most of the requirements described in section I, but its rater credibility model has its limitation as it is based on the assumption that the majority of the rater is honest, if this assumption is not true then the whole system fails. [11] applies a fuzzy method to investigate the relationship between user rating and QoS compliance, to infer the rationale for ratings and identify user preferences. Yet no metrics is given to derive web services' final trust, hence the consumer can not select the web service after all. [7] and [10] also attempt to combine user rating and QoS monitoring; but they do not address the multi-QoS and the user preference problems, and no final trust value is shown.

Different to the above approaches, a few models use other information to compute the trust value. For example, [4] uses the association coefficient between web services, i.e. by interacting with well-reputed web services, the target web service can improve its reputation. Although these models provide some interesting ideas, they can not solve the user preferences and the multi-QoS problems.

Moreover, the most common limitation of current approaches is that they do not clarify clearly the mechanism for user rating. All solutions imply that users provide ratings for web services. A more comprehensive solution should consider it to be the rating for a particular transaction and then aggregate all transactions to get the feedback for the service because a user can consume a service several times. Also, most of them do not specify the form of ratings (a score, a level or a binary value?)

The methods for calculating rater credibility in most of the existing models are not fine enough either. For instance: in [2] credibility of a rater equals to his own reputation. This causes an "infinite loop" in the formula. In [5], a rater is assessed based on the number of feedback he provided, which is unreasonable because the quantity can not reflect the quality correctly. The authors in [8] propose a very complicated method to calculate rater credibility; however, it faces a serious problem when dishonest raters form the majority in the community. It assumes that most raters provide fair ratings and filter the false rating based on its similarity with the majority.

Our approach tries to overcome those limitations by:

1) Integrating three different kinds of trust sources: user rating, QoS monitoring and direct experience of the requester in order to have both subjective and objective view of web service trust.
2) Allowing users to specify their QoS preferences.
3) Clarifying the rating mechanisms: consumer gives a score for each quality attribute after each transaction.
4) Evaluating the credibility of a rater based on the usefulness of the rater' feedback and the similarity between the rater and the requester.
5) Applying Bayesian network method. Actually, not many current web service trust models apply Bayesian network. In contrast, Bayesian methods have been used in many trust and reputation models for other systems for example [13], but most of those models are ad-hoc because they apply complex beta distribution. They only allow binary ratings which are very simple and inappropriate to capture the ranking of web services, and moreover, they do not address the problem of multi-context, personalized trust. Our model is inspired by the work in [12], which presents a trust model for P2P systems; however, we improve it to handle multiple levels of quality attributes and QoS performance conformance for the web service systems.

III. The Trust and Reputation Model

In our model, three kinds of sources: the direct experience opinion, the recommendation from other consumers, and the

QoS monitoring information are weighted to derive the final reputation of the web service:

$$T_x(i) = T_{dx}(i) \times \omega_d + T_{rx}(i) \times \omega_r + T_{cx}(i) \times \omega_c \quad (1)$$

$T_x(i)$ is the final reputation value of web service i, in the view of consumer x (the truster). $T_{dx}(i)$, $T_{rx}(i)$ and $T_{cx}(i)$ called direct trust, recommendation trust and conformance trust are the reputation values of web service i obtained by the direct experience mechanism, the recommendation mechanism and the QoS monitoring mechanism. ω_d, ω_r and ω_c are weighing factors of these values. $\omega_d + \omega_r + \omega_c = 1$. In the following sub-sections we will clarify how to compute these three reputation values.

A. Trust based on direct experience

We propose a Bayesian network model to compute trust values. The idea behind that comes from the problem of multiple QoS context trust. In reality, different service domain has different set of QoS and different service consumer is also interested in different QoS and when selecting a web service, different consumer also wants to know the reputation of the web service in different QoS. For example, in a SMS web service, a consumer may want to know how fast and how reliable the web service is, i.e. is interested in the time performance and reliability, while in email verification or currency exchange rates web services, the consumers may be interested in the accuracy.

In this context, a Bayesian network is an appropriate model because it is a modern statistic method to calculate the probability of a hypothesis under different conditions that are equivalent to different quality attributes of the web service. The theoretical background of Bayesian network is the rule:

$$P(h|c) = \big(P(c|h) \times P(h)\big)/P(c) \quad (2)$$

$$P(h|c) \times P(c) = P(c|h) \times P(h) \quad (3)$$

In the above equations, $P(h|c)$ is the probability of the hypothesis h given the condition or the evidence c; $P(c|h)$ is the probability of c given h. $P(h)$ is the probability of hypothesis h in general; $P(c)$ is the probability of condition or evidence c.

In the Bayesian network trust model for web service, the root node (S) is the satisfaction in overall capability of the web service. S can be either 1 (*satisfactory*) or 0 (*unsatisfactory*). Each leaf node $Lq_j (j \in [1-n])$ represents the level for the quality j. Its values set is the set {1, 2, 3, 4, 5}, which means {"*very bad*", "*bad*", "*satisfactory*", "*good*", "*excellent*"}. The trustworthy or reputation of a web service i in the aspect of quality j at a level a, assessed by consumer x can be seen as the probability that consumer x is satisfied with the web service i in a transaction given that quality j must be at least at level a, times the probability that quality j is at least at level a. So we need to compute:

$$\begin{aligned} T_{dx}(i, Lq_j \geq a) &= P_{x,i}(S = 1|Rq_j \geq a) \times P_{x,i}(Rq_j \geq a) \\ &= P_{x,i}(Rq_j \geq a|S = 1) \times P_{x,i}(S = 1) \end{aligned} \quad (4)$$

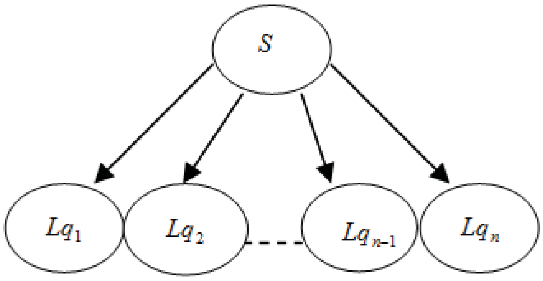

Figure 1. Bayesian Network of Web Service Trust

Each consumer builds a Bayesian network for each web service that it has interacted with by maintaining a set of conditional probability tables (CPT) for that web service. Each CPT is for a quality attribute of that service. When requesting for the service, the consumer will specify its level of interest Wq_j {0, 1, 2}, which means {"*not interested*", "*interested*", "*very interested*"} for each quality j of service i. After each transaction, consumer x will provide the rating score $Rq_j(x, i, u) \in [1-5]$ for each quality j of service i; based on that, the overall rating $R(x, i, u)$ that consumer x gives to web service i in transaction u will be calculated as follows:

$$R(x, i, u) = \frac{TF(x, i, u) \times \sum_{j=1}^{n} \big(Rq_j(x, i, u) \times Wq_j(x, i)\big)}{\sum_{j=1}^{n} Wq_j(x, i)} \quad (5)$$

$TF(x, i, u) \in [0-1]$ is the transaction context factor of this transaction. To assess a web service, we also should consider characteristics of the service and the transactions such as transaction time, service category etc. For example, a service doing critical tasks should be weighted more than some simple trivial web services, or if the quotation is much (the consumer spends large amount) and the transaction shows good results, it can boost the web service's reputation more. Also we should decay very old transactions and feedbacks. Thus we need a function to integrate these factors, for instance:

$$TF(x, i, u) = f\big(category(i), cost(u), time(u)\big) \quad (6)$$

However at the moment we do not focus on this matter but reserve it for future work.

Formula (5) makes $R(x, i, u)$ in the range $[1-5]$. Each consumer x defines a threshold Th_x for satisfaction. If $R(x, i, u) \geq Th_x$, the web service is satisfactory in this transaction, which means $S = 1$ otherwise $S = 0$.

The CPT for each quality will be updated by filing the table with the values $P_{x,i}(Rq_j = a|S = b)$; in which $a \in$

$[1-5]; b \in \{0,1\}$. This is the probability that quality j is rated at level a given that the transaction is satisfactory (for $b=1$) or unsatisfactory (for $b=0$).

$$P_{x,i}(Rq_j = a|S=1) = \frac{N(x,i,Rq_j=a,S=1)}{N(x,i,S=1)} \quad (7)$$

$N(x,i,S=1)$ is the number of satisfactory interaction between consumer x and web service i. $N(x,i,Rq_j = a, S=1)$ is the number of satisfactory interaction between consumer x and web service i in which quality j is rated at level a.

Table I
THE CPT OF NODE QUALITY j

	$S=1$	$S=0$
$Rq_j = 1$ very bad	$P(Rq_j = 1\|S=1)$	$P(Rq_j = 1\|S=0)$
$Rq_j = 2$ bad	$P(Rq_j = 2\|S=1)$	$P(Rq_j = 2\|S=0)$
$Rq_j = 3$ satisfactory	$P(Rq_j = 3\|S=1)$	$P(Rq_j = 3\|S=0)$
$Rq_j = 4$ good	$P(Rq_j = 4\|S=1)$	$P(Rq_j = 4\|S=0)$
$Rq_j = 5$ excellent	$P(Rq_j = 5\|S=1)$	$P(Rq_j = 5\|S=0)$

From these table values we can compute:

$$P_{x,i}(Rq_j \geq a|S=1) = \sum_{k=a}^{5} P_{x,i}(Rq_j = k|S=1) \quad (8)$$

We can also compute $P_{x,i}(S=1)$ - the percentage of satisfactory transaction between consumer x and web service i, measured by the number of satisfactory transactions divided by the total number of transactions between the two:

$$P_{x,i}(S=1) = N(x,i,S=1)/N(x,i) \quad (9)$$

So we can plug the calculated values from equation (8), (9) into equation (4) to obtain the value for direct experience trust of a consumer x in web service i . The advantage is that consumer x can assess the reputation of web service i in the aspect of many qualities at the same time. For example consumer x wants to know how it can trust web service i in providing excellent performance ($Lq_1 \geq 5$) and good price ($Lq_2 \geq 4$):

$$\begin{aligned} T_{dx}(i, Lq_1 \geq 5, Lq_2 \geq 4) &= P_{x,i}(S=1|Rq_1 \geq 5, Rq_2 \geq 4) \\ &\quad \times P_{x,i}(Rq_1 \geq 5) \times P_{x,i}(Rq_2 \geq 4) \\ = P_{x,i}(Rq_1 \geq 5|S=1) &\times P_{x,i}(Rq_2 \geq 4|S=1) \times P_{x,i}(S=1) \end{aligned} \quad (10)$$

B. Trust based on recommendation

The essences of direct trust $T_{dx}(i)$ and recommendation trust $T_{rx}(i)$ are the same because they both come from the ratings of service consumers for the target web service. The only difference is that in direct trust, the subject of the rating is the requester itself whereas in recommendation trust, the ratings come from a third party. Hence we can apply a same method to calculate these two reputation values, but in recommendation trust, we should consider also rater credibility, or the trust in the service consumer's ability to provide fair recommendations.

The recommendation trust $T_{rx}(i)$ for web service i, in the view of consumer x, is the integration of all direct trust for web service i from all consumers except for x that has interacted with the web service, considering the credibility of the consumers.

$$T_{rx}(i) = \sum_{\substack{y \in CS(i) \\ y \neq x}} \big(T_{dy}(i) \times Cr_x(y)\big) \Big/ \sum_{\substack{y \in CS(i) \\ y \neq x}} Cr_x(y) \quad (11)$$

$Cr_x(y)$ is the credibility of consumer y as a rater, in the view of consumer x. We consider two factors to give rater y credit: (i) the usefulness of rater y's feedbacks, (ii) the similarity between the ratings and preferences of consumer x and those of rater y. The credibility of rater y is the weighted sum of these two factors:

$$Cr_x(y) = Uf(y) \times \omega_{uf} + Si(x,y) \times \omega_{si} \quad \omega_{uf} + \omega_{si} = 1 \quad (12)$$

By this way, consumer x can evaluate rater y in both a personalized way (the similarity measure) and a public way (the usefulness measure) but without much dependence on other raters (we do not compare with the majority ratings or some special agents'ratings).

The usefulness of rater y's feedbacks:

$$Uf(y) = N_{uf}(y)/N_f(y) \quad (13)$$

$N_{uf}(y)$ is the number of useful feedbacks from rater y, $N_f(y)$ is the total number of feedbacks from rater y. To assess if the feedback is useful or not, we use these steps:

1. When consumer x requests the system for the recommendation trust of service i, the system queries the target raters for the feedbacks. If y provides feedback $T_{dy}(i)$ about service i and after the trust assessment process, consumer x selects and consumes service i in transaction u, the system will keep the record $(Id_u, T_{dy}(i), Id_x, Id_y)$ and increase $N_f(y)$. If consumer x does not select service i then $T_{dy}(i)$ is simply discarded.

2. After consumer x interacts with service i in transaction u, x will report the satisfaction $S(u)=1$ or $S(u)=0$.

3. If $T_{dy}(i) > Th \wedge S(u) = 1$ or $T_{dy}(i) < Th \wedge S(u) = 0$ (Th is the threshold defined by the system) then the feedback from rater y is said to be useful and $N_{uf}(y)$ is increased, otherwise it is not useful.

The similarity between consumer x and rater y:

$$Si(x,y) = 1 - D(x,y) \quad (14)$$

The distance $D(x,y)$ between consumer x and rater y includes two parts: the distance between the ratings and the distance between QoS preferences:

$$D(x,y) = \big(D_r(x,y) + D_p(x,y)\big)\big/2 \quad (15)$$

These two distance values can be calculated based on the Euclidean method. Let $SE(x,y)$ be the set of all mutual services that both x and y have interacted with and provided feedbacks to, $Wq_j(x,i)$ be the level of interest of consumer x in quality q_j of service i, $\overline{R}q_j(x,i)$ denote the average rating score that consumer x gives to quality j of service i over many transactions. Then:

$$\overline{R}q_j(x,i) = \Big(\sum_{u=1}^{N(x,i)} Rq_j(x,i,u)\Big) \Big/ N(x,i) \tag{16}$$

$$D_r(x,y) = \frac{\displaystyle\sum_{i \in SE(x,y)} \sqrt{\frac{\sum_{j=1}^{n_i} \left(\overline{R}q_j(x,i) - \overline{R}q_j(y,i)\right)^2 \times Wq_j(x,i)}{\sum_{j=1}^{n_i} Wq_j(x,i)}}}{4 \times |SE(x,y)|} \tag{17}$$

$$D_p(x,y) = \frac{\displaystyle\sum_{i \in SE(x,y)} \sqrt{\frac{\sum_{j=1}^{n_i} \left(Wq_j(x,i) - Wq_j(y,i)\right)^2 \times Wq_j(x,i)}{\sum_{j=1}^{n_i} Wq_j(x,i)}}}{2 \times |SE(x,y)|} \tag{18}$$

$D_r(x,y)$ and $D_p(x,y)$ are in the range $[0-1]$ and thus $Si(x,y)$ and $Cr_x(y)$ are also in the range $[0-1]$.

C. Trust based on QoS conformance

The idea is to determine the conformance level of each quality of the service by comparing the delivered QoS values and the promised or projected QoS values. Suppose the system can measure the delivered value q_{dj} of quality j of service after each invocation and the projected QoS value q_{pj} is stored in the registry at the beginning. Let $C_j \in [-1,1]$ be the compliance value of quality j of service i in a transaction. The higher value of C_j is, the better service performance is. If $dir(q_j) \uparrow$ (the higher value of q_j , the better), then we want $q_{dj} \geq q_{pj}$, hence

$$C_j = (q_{dj} - q_{pj})/q_{pj} \tag{19}$$

If $dir(q_j) \downarrow$ (the lower value of q_j , the better), then we want $q_{dj} \leq q_{pj}$, hence

$$C_j = (q_{pj} - q_{dj})/q_{pj} \tag{20}$$

We use the transformation below to indentify *conformance level* Cq_j of q_j based on the values of C_j

$$Cq_j = \begin{cases} 1 \text{ (very bad)} & \text{if } -1 \leq C_j < -0.5 \\ 2 \text{ (bad)} & \text{if } -0.5 \leq C_j < 0 \\ 3 \text{ (satisfactory)} & \text{if } C_j = 0 \\ 4 \text{ (good)} & \text{if } 0 < C_j < 0.5 \\ 5 \text{ (excellent)} & \text{if } 0.5 \leq C_j < 1 \end{cases}$$

The aggregated compliance level of service i over many QoS in a transaction u in the view of consumer x is:

$$C_u = \frac{\sum_{j=1}^{n_i} Cq_j \times Wq_j(x,i)}{\sum_{j=1}^{n_i} Wq_j(x,i)} \tag{21}$$

If $C_u \geq 3$, the service's conformance in this transaction is overall satisfactory ($S_c = 1$), otherwise it is unsatisfactory ($S_c = 0$). Now we can use the similar Bayesian network model as in section III-A to derive the conformance trust for service i, but the rating Rq_j is replaced by the conformance Cq_j .The system keeps the CPT for each quality of each service (similar to table I) and updates it after each transaction.

So when the consumer wants to assess the reputation of web service i in the aspect of some target qualities, the Bayesian models will also be applied for conformance trust $T_{cx}(i)$. For example if consumer x wants to know how it can trust web service i in providing excellent performance ($Lq_1 \geq 5$) and good price ($Lq_2 \geq 4$) , then we should calculate

$$\begin{aligned} T_{cx}(i, Lq_1 \geq 5, Lq_2 \geq 4) &= P_{ci}(S_c = 1 | Cq_1 \geq 5, Cq_2 \geq 4) \\ &\quad \times P_{ci}(Cq_1 \geq 5) \times P_{ci}(Cq_2 \geq 4) \\ = P_{ci}(Cq_1 \geq 5 | S_c = 1) &\times P_{ci}(Cq_2 \geq 4 | S_c = 1) \times P_{ci}(S_c = 1) \end{aligned} \tag{22}$$

IV. Algorithms

When consumer x asks the system for the best service, x supplies the system with a set of requirements: $Re = \big\{Re_j = \{q_j, l_j, Wq_j\}\big\}$. $q_j$is the ID of a quality attribute that belongs to a set of standard attributes defined by the system, for example ”response time”, ”availability”, ”cost” etc. $l_j \in [1-5]$ is the required level of q_j , Wq_j is the level of interest of the user in quality j. These inputs are used to calculate the trust of candidate web services. The detail of the calculation phases are presented in the following algorithms:

Algorithm 1 describes the system's overall process to compute the trust of each web service in the candidate list.

Algorithm 2 describes what consumer x does after each transaction with a web service i and what the system does as responses.

Algorithm 3 shows the steps to calculate the direct trust of a web service i in the view of consumer x based on its requirements.

Algorithm 4 presents the method to compute the credibility of a rater y, in the view of consumer x.

Algorithm 5 shows the process to calcualte the conformance trust of a web service i in the view of consumer x based on its requirements.

```
record ID of $Re$, ID of $x$;
for each $WS_i \in WS$ (list of candidate web services) do
    set trust values $T_x(i) = 0$; $T_{dx}(i) = 0$; $T_{rx}(i) = 0$; $T_{cx}(i) = 0$;
    ask $x$ for $T_{dx}(i)$ ; get consumers list of $WS_i : CS_i = \{y\}$;
    set $A = 0$ ; $B = 0$;
    for each consumer $y \in CS_i$ do
        ask $y$ for direct trust $T_{dy}(i)$;
        record $ID_y$ , $T_{dy}(i)$ in the rater list $R_{re}$ for this request;
        calculate $Cr_x(y)$;
        $A = A + T_{dy}(i) \times Cr_x(y)$; $B = B + Cr_x(y)$;
    end
    set $T_{rx}(i) = A/B$; compute conformance trust $T_{cx}(i)$;
    set $T_x(i) = T_{dx}(i) \times \omega_d + T_{rx}(i) \times \omega_r + T_{cx}(i) \times \omega_c$;
end
return the list $WS$ sorted in descending order by $T_x(i)$;
```

Algorithm 1: CalculateWSReputation

```
$x$ gives ratings for web service $i$;
$x$ calculates and submits the satisfaction of transaction $S \in \{0,1\}$;
$x$ records ID of transaction $u$, ID of $i$, rating of each quality $Rq_j$;
$x$ updates values of $N(x,i,S=1)$, $N(x,i,S=0)$, $N(x,i)$;
for each quality $q_j$ do
    $x$ updates CPT for $q_j$;
end
if $x \notin CS_i$ (Consumer list of i) then
    add $x$ to $CS_i$;
end
derive the list of rater $R_{re}$ for $x$'s request;
for each rater $y \in R_{re}$ do
    $N_f(y) = N_f(y) + 1$;
    if ($T_{dy}(i) > Th$ && $S(u) = 1$) or ($T_{dy}(i) < Th$ && $S(u) = 0$) then
        update $N_{uf}(y) = N_{uf}(y) + 1$;
    end
end
measure and record delivered values of each quality $q_{dj}$;
return
```

Algorithm 2: UpdateAfterTransaction$\Big($Consumer x, WebService i, TransactionFactor TF, QoSRequirements $Re = \Big\{Re_j = \{q_j, l_j, Wq_j\}\Big\}\Big)$

```
if $N(x,i) \neq 0$ && $\exists CPT(x,i)$ then
    $P_{x,i}(S=1) = N(x,i,S=1)/N(x,y)$; $T_{dx}(i) = P_{x,i}(S=1)$;
    for each $Re_j \in Re$ do
        if $l_j \neq 1$ then
            use CPT to calculate $P_{x,i}(Rq_j \geq l_j | S=1)$;
            $T_{dx}(i) = T_{dx}(i) \times P_{x,i}(Rq_j \geq l_j | S=1)$;
        end
    end
else
    $T_{dx}(i) = 0$;
end
return $T_{dx}(i)$;
```

Algorithm 3: CalculateDirectTrust$\Big($Consumer x, WebService i, QoSRequirements $Re = \Big\{Re_j = \{q_j, l_j, Wq_j\}\Big\}\Big)$

```
retrieve values of $N_{uf}(y)$ and $N_f(y)$;
if $N_f(y) = 0$ then
    $Uf(y) = 0$;
else
    $Uf(y) = N_{uf}(y)/N_f(y)$;
end
find $SE(x,y)$ - the set of mutual web services of $x$ and $y$;
set $D_r(x,y) = 0$; $D_p(x,y) = 0$;
for each web service $i \in SE(x,y)$ do
    set $A = 0$; $B = 0$; $C = 0$;
    retrieve the level of interest of $x$ and $y$ in QoS of service $i$;
    for each quality $q_j$ of service $i$ do
        calculate average rating: $\overline{R}q_j(x,i)$, $\overline{R}q_j(y,i)$;
        set $A = A + \Big(\overline{R}q_j(x,i) - \overline{R}q_j(y,i)\Big)^2 \times Wq_j(x,i)$;
        set $B = B + \Big(Wq_j(x,i) - Wq_j(y,i)\Big)^2 \times Wq_j(x,i)$;
        set $C = C + Wq_j(x,i)$;
    end
    $D_r(x,y) = D_r(x,y) + \sqrt{A/C}$;
    $D_p(x,y) = D_p(x,y) + \sqrt{B/C}$;
end
set $D_r(x,y) = D_r(x,y)/\Big(4 \times |SE(x,y)|\Big)$;
set $D_p(x,y) = D_p(x,y)/\Big(2 \times |SE(x,y)|\Big)$;
set $D(x,y) = \Big(D_r(x,y) + D_p(x,y)\Big)/2$;
$Si(x,y) = 1 - D(x,y)$; $Cr_x(y) = Uf(y) \times \omega_{uf} + Si(x,y) \times \omega_{si}$;
return $Cr_x(y)$;
```

Algorithm 4: CalculateRaterCredibility$\Big($Consumer x, Consumer $y\Big)$

```
for each transacion $u$ of $i$ do
    set $A = 0$; $B = 0$;
    for each quality $q_j$ of service $i$ do
        get projected value $q_{pj}$ & delivered value $q_{dj}$ in transaction $u$;
        if $dir(q_j) \uparrow$ then
            $C_j = (q_{dj} - q_{pj})/q_{pj}$;
        else
            $C_j = (q_{pj} - q_{dj})/q_{pj}$;
        end
        derive value of $Cq_j[1-5]$ from $C_j$;
        set $A = A + Cq_j \times Wq_j$; $B = B + Wq_j$;
    end
    calculate compliance level of service $i$ in $u$: $C_u = A/B$;
    if $C_u \geq 3$ then
        satisfaction of compliance $S_c = 1$;
    else
        $S_c = 0$;
    end
    calculate $P_{ci}(S_c = 1) = N(i, S_c = 1)/N(i)$;
    set $T_{cx}(i) = P_{ci}(S_c = 1)$;
    for each $Re_j \in Re$ do
        $T_{cx}(i) = P_{ci}(Cq_j \geq l_j | S_c = 1) \times T_{cx}(i)$;
    end
    return $T_{cx}(i)$;
end
```

Algorithm 5: CalculateConformanceTrust$\Big($Consumer x, WebService i, QoSRequirements $Re = \Big\{Re_j = \{q_j, l_j, Wq_j\}\Big\}\Big)$

V. An Example

We show an example of how the system evaluates the trust of web services based on the consumers preference. Suppose there are three consumers CS1, CS2, and CS3, who have used the web services of same functionality in the set: WS = {WS1, WS2, WS3, WS4, WS5}. Our example focuses on the three typical QoS attributes: Response Time (q_r), availability (q_a), cost (q_c). We know that $dir(q_r) \downarrow$, $dir(q_a) \uparrow$ and $dir(q_c) \downarrow$ .The profile of each consumer is shown in table II, with the level of interest $[0-2]$ of each consumer in each quality attribute. Each consumer focuses on a different target QoS attribute. For simplicity we assume that the system has calculated the usefulness of each consumer's feedback beforehand. Consumer 3 is a dishonest rater.

Table II
Consumers' profiles

	Wq_r	Wq_a	Wq_c	Uf
CS1	2	1	1	0.8
CS2	1	2	1	0.8
CS3	1	1	2	0.5

Table III
Web Services' profiles

	Response Time		Availability (%)		Cost	
	Actual	q_{pr}	Actual	q_{pa}	Actual	q_{pc}
WS1	G	35	B	50	G	20
WS2	B	65	G	90	G	30
WS3	S	40	G	90	B	70
WS4	S	50	S	60	G	20
WS5	G	20	S	70	S	40

Table III shows the web services profiles with the projected values of quality attributes and their actual performance (the system does not know about this actual performance). The values of quality attributes are in $[0-100]$. For a given quality q_j: (i) if $q_j > 70$, if $dir(q_j) \uparrow$ it is considered to be at good level (G) but if $dir(q_j) \downarrow$, it is at bad level (B); (ii) if $q_j < 40$, if $dir(q_j) \uparrow$ it is at B but if $dir(q_j) \downarrow$, it is at G; (iii) if $40 < q_j < 70$, it is at satisfactory level (S).

The interactions histories of consumers are shown in tables IV, V, and VI. Rq_j and q_{dj} are the rating and the actual delivered value of quality q_j in the interactions.

Now the question is that consumer 1 wants to know how it can trust those web services in providing satisfactory performance ($Lq_r \geq 3$) and satisfactory level of availability ($Lq_a \geq 3$). The system will evaluate the final trust of web service i ($i \in [1-5]$) in the view of CS1:

$$T_1(i, Lq_r \geq 3, Lq_a \geq 3) = T_{d1}(i, Lq_r \geq 3, Lq_a \geq 3) \times \omega_d + T_{r1}(i, Lq_r \geq 3, Lq_a \geq 3) \times \omega_r + T_{c1}(i, Lq_r \geq 3, Lq_a \geq 3) \times \omega_c$$

In this example, $\omega_d = 0.4$, $\omega_r = 0.25$, $\omega_c = 0.35$

Consumers will calculate S - the overall satisfaction in the interactions. The QoS monitoring engine will measure the delivered quality values and calculate Cq_j - the conformance level of each quality in the interaction and S_{c1} - the overall satisfaction about the conformance level in the view of CS1. To make it simple, we assume that $TF = 1$ for all transactions. The threshold for $S = 1$ is that the overall rating for the transaction $R \geq 3.5$; the threshold for $S_{c1} = 1$ is that the overall conformance level $C \geq 3$. There are four steps in the computation:

Step 1: Each consumer applies Bayesian network to calculate its direct trust for the web services (algorithm 3).

Step 2: The system calculates the credibility of the raters (CS2 and CS3) to derive the recommendation trust of each web service, in the view of CS1 (algorithm 4 and 1).

Step 3: The system calculates the conformance trust of each web service in the view of CS1 (algorithm 5).

Step 4: Final trust value is computed (algorithm 1).

Table IV
TRANSACTION HISTORY OF CONSUMER 1

CS1	Response Time			Availability (%)			Cost			S	S_{c1}
	Rq_r	q_{dr}	Cq_r	Rq_a	q_{da}	Cq_a	Rq_c	q_{dc}	Cq_c		
WS1											
T1	4	30	4	2	40	2	4	30	2	1	1
T2	4	25	4	1	30	2	4	25	2	0	1
T3	3	40	2	2	50	3	4	30	2	0	0
Ave	3.67			1.67			4				
WS2											
T1	4	40	3	5	90	3	3	70	3	1	1
T2	4	40	3	4	80	2	1	85	2	0	0
T3	2	60	2	5	95	4	3	70	3	0	0
Ave	3.33			4.67			2.33				
WS5											
T1	4	25	2	3	75	4	4	35	4	1	1
T2	5	15	4	4	75	4	3	50	2	1	1
T3	4	20	3	3	70	3	2	60	2	0	0
Ave	4.33			3.33			3				

Table V
TRANSACTION HISTORY OF CONSUMER 2

CS2	Response Time			Availability (%)			Cost			S	S_{c1}
	Rq_r	q_{dr}	Cq_r	Rq_a	q_{da}	Cq_a	Rq_c	q_{dc}	Cq_c		
WS2											
T1	2	65	3	5	95	4	3	30	3	1	1
T2	2	60	4	3	75	2	3	35	2	0	1
T3	3	70	2	4	90	3	3	40	2	1	0
Ave	2.33			4			3				
WS3											
T1	3	40	3	4	90	3	2	70	3	0	1
T2	2	40	3	4	95	4	3	60	4	0	1
T3	3	45	2	3	80	2	2	75	2	0	0
T4	4	30	4	4	85	2	3	70	3	1	1
Ave	3			3.75			2.5				
WS4											
T1	3	45	4	3	60	3	4	25	2	0	1
T2	4	40	4	2	40	2	4	20	3	0	1
T3	3	55	2	3	50	2	5	15	4	1	0
Ave	3.33			2.67			4.33				

Table VI
TRANSACTION HISTORY OF CONSUMER 3

CS3	Response Time			Availability (%)			Cost			S	S_{c1}
	Rq_r	q_{dr}	Cq_r	Rq_a	q_{da}	Cq_a	Rq_c	q_{dc}	Cq_c		
WS1											
T1	4	35	3	3	50	3	4	20	3	1	1
T2	4	30	4	2	40	2	4	20	3	1	1
T3	4	25	4	2	45	2	4	30	2	1	1
Ave	4			2.33			4				
WS2											
T1	2	70	2	5	90	3	3	40	2	0	0
T2	2	65	3	4	80	2	4	30	3	1	0
T3	2	60	4	4	80	2	4	35	2	1	1
Ave	2			4.33			3.67				
WS3											
T1	4	40	3	4	90	3	2	70	3	0	1
T2	3	45	2	5	95	4	3	65	4	1	1
Ave	3.5			4.5			2.5				
WS5											
T1	4	25	2	3	70	3	2	60	2	0	0
T2	4	30	2	3	65	2	3	40	3	0	0
T3	5	15	4	4	75	4	4	35	4	1	1
Ave	4.33			3.33			3				

The results are summarized in table VII, which includes the direct trust of each consumer CS1, CS2, CS3 in each web service, the conformance trust, the recommendation trust and the final trust of those web services in the view of consumer CS1.

Table VII
SUMMARY OF COMPUTED TRUST VALUES

	$T_{d1}(i)$	$T_{d2}(i)$	$T_{d3}(i)$	$T_{r1}(i)$	$T_{c1}(i)$	$T_1(i)$
WS1	0	N/A	1/3	1/3	1/6	**0.142**
WS2	N/A	1/3	0	0.146	1/6	**0.095**
WS3	1/3	1/4	1/2	0.39	0.463	**0.393**
WS4	N/A	1/3	N/A	1/3	1/3	**0.2**
WS5	2/3	N/A	1/3	1/3	1/3	**0.467**

Based on the final trust values, the order of the web services that consumer 1 should trust in providing satisfactory level of time performance and availability are WS5, WS3, WS4, WS1, WS2. This conforms to the fact shown in table 2 about the actual performance of the web services. Since CS1 focuses on response time the most, then availability, WS5 with actual QoS levels of GSS should have the highest trust value then come WS3 and WS4. WS1 and WS2 both have two good quality levels but since response time weights more than availability, the final computed trust values should reflect that WS1 is better than WS2. Also, although CS1 has no experience in interacting with WS4 (direct trust is zero), WS4's final trust value is still higher than WS1's. This is reasonable because both actual levels of response time and availability of WS4 are satisfactory while the actual availability level of WS1 is bad. This shows the importance of recommendation trust and conformance trust besides direct trust.

In summary, this example shows that the proposed model can provide sound results to assess the trust and reputation of web services.

VI. Conclusion

We have presented a Bayesian network trust and reputation model for web services. Our approach overcomes some of the limitations of existing models by combining different trust sources in both subjective view and objective view; focusing on satisfying users preferences on different QoS attributes and assessing rater credibility in an appropriate way. Our example scenario shows that the proposed model can provide good results so that the consumer can select the best web service that suits its requirements.

At this stage we have not focused on some important issues for example the transaction context factors, the bootstrapping problem, the design and implementation issues, unfair ratings, incentive for rating. We are working on these issues to build a comprehensive framework for assessing trust and reputation of web services.

References

[1] E. M. Maximilien and M. P. Singh. Multiagent System for Dynamic Web Services Selection. In *Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS)*, Utrecht, the Netherlands, July 25-29, 2005.

[2] K. Karta. An Investigation on Personalized Collaborative Filtering for Web Service Selection. *Technical report*, Available online (last accessed on May. 15, 2009).

[3] U. S. Manikrao and T. V. Prabhakar. Dynamic Selection of Web Services with Recommendation System. *International Conference on Next Generation Web Services Practices*, Seoul, Korea, August 2005.

[4] P. K. Atrey, M. A. Hossain and A. E. Saddik. Association based dynamic computation of reputation in web services. *Journal of Web and Grid Services*, vol. 4, no. 2, 2008.

[5] J. H. Abawajy and A. M. Goscinski. A Reputation-Based Grid Information Service, In *International Conference on Computational Science (ICCS'06)*, vol 3994 of *Lecture Notes in Computer Science*, pages 10151022. Springer, 2006.

[6] L. H. Vu, M. Hauswirth and K. Aberer. QoS-based service selection and ranking with trust and reputation management, In *Proceedings of OTM'05*, R. Meersman and Z.Tari (Eds.), LNCS 3760, p.p. 466-483, 2005.

[7] L. Tan, C-H. Chi and J. Deng. Quantifying trust based on service level agreement for software as a service. In *Proceedings of IEEE International Computer Software and Applications Conference*, 2008.

[8] Z. Malik and A. Bouguettaya. Rater credibility assessment in web services interactions. *World Wide Web Journal*, vol. 12, no. 1, March 2009.

[9] S. Kalepu, S. Krishnaswamy and S. W. Loke. Reputation = f(User Ranking, Compliance, Verity). In *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, 2004.

[10] G. Spanoudakis and S. LoPresti. Web service trust: Towards a dynamic assessment framework. In *Proceedings of the International Conference on Availability, Reliability and Security*, 2009.

[11] W. Sherchan, S. W. Loke and S. Krishnaswamy. A fuzzy model for reasoning about reputation in web services. In *Proceedings of ACM Symposium on Applied Computing*, New York, p.1886, ACM Press, 2006.

[12] Y. Wang and J. Vassileva. Bayesian network-based trust model. In *Proceedings of the IEEE International Conference on Web Intelligence (WI'03)*, 2003.

[13] A. Josang. The beta reputation system. In *Proceedings of 15th Bled Electronic Commerce Conference*, Bled, Slovenia, June 17-19, 2002.